

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Dizertační práce

Metody sběru a zpracování dat v prostředí WWW

Autor: Ing. Jan Masner

Obor: Informační management

© 2018 ČZU v Praze

Poděkování

Rád bych poděkoval in memoriam svému školiteli doc. Ing. Zdeňku Havlíčkovi, CSc. za odborné vedení, cenné rady a podněty při zpracovávání podstatné části této práce.

Dále pak děkuji svým kolegům z Katedry informačních technologií za připomínky a rady při dokončování práce.

V neposlední řadě bych rád poděkoval současným i bývalým členům redakce studentských univerzitních novin iZUN.eu, zejména Ing. Jiřímu Opletalovi, který přišel se základní myšlenkou a stál se mnou u zrodu celého projektu.

Metody sběru a zpracování dat v prostředí WWW

Abstrakt

Předkládaná dizertační práce se tematicky zaměřuje na problematiku informačního obsahu v prostředí World Wide Web. Hlavním cílem je navrhnout metodiku pro vkládání, aktualizaci, uložení a prezentaci informačního obsahu v prostředí WWW. Na základě analýzy stavu poznání v oblasti data, informace, znalosti, je vymezen pojem informační obsah vzhledem k zaměření práce. Dále jsou analyzovány současné webové technologie, sémantika a redakční systémy jako nástroje pro sběr a zpracování dat v prostředí WWW. Blíže jsou zkoumány možnosti, které redakční systémy a jejich rozšíření nabízejí pro správu informačního obsahu. Syntézou poznatků z provedených analýz je stanovena hypotéza dizertační práce a v návaznosti také cíle a požadavky na navrhovanou metodiku, která je pro účely práce pojmenována WICM (Web Information Content Management). Návrh metodiky je tvořen obdobným způsobem, jako současné webové standardy a specifikace. V textu je rozdělen do tří částí: Vkládání a aktualizace obsahu (1), Ukládání obsahu (2), Prezentace, sémantika a archivace (3). Uživatelská část metodiky je následně ověřena pomocí metody Focus Groups. Celkově pak je metodika ověřena pomocí série polo-strukturovaných rozhovorů s odborníky z praxe. Na závěr práce jsou zhodnoceny přínosy a možnosti pro další výzkum v dané oblasti.

Klíčová slova

WWW, HTML, internet, redakční systémy, sémantický web, metadata, informační obsah

Methods of collecting and processing of data in WWW environment

Abstract

This dissertation thesis is focused on the information content in World Wide Web environment. The main objective is to design a methodology for inserting, updating, storage and presentation of information content in WWW. At first, the term “information content” is defined on the basis of *data, information and knowledge* analysis. Furthermore, there is an analysis of current web technologies, semantics and content management systems as a tool for data collection and processing in WWW. Possibilities of CMS and their extensions for management of the information content are studied. Gained knowledge is synthesized and subsequently, the hypothesis of the dissertation is assessed as well as objectives and requirements for the proposed WICM methodology. The methodology is designed in a similar way as current web standards and specifications. It is divided into three parts: Inserting an updating of a content (i), Content storage (ii), Presentation, semantics and archiving (iii). A part of the methodology (desired for users) is validated using Focus Groups method. Alongside, semi-structured interviews are used for overall validation. Contributions of the methodology and possibilities for subsequent research are assessed in conclusion.

Keywords

WWW, HTML, internet, content management systems, semantic web, metadata, information content

Obsah

1	Úvod	4
2	Cíle práce	6
2.1	Hlavní cíl.....	6
2.2	Dílčí cíle dizertační práce.....	6
3	Metodický postup	7
3.1	Metodický aparát	9
3.1.1	Metodika.....	9
3.1.2	Metoda.....	9
3.1.3	Prototypový přístup.....	9
3.1.4	Metoda Focus Groups.....	10
3.1.5	UML a doménový model.....	10
4	Literární přehled	12
4.1	Data, informace, znalosti	12
4.1.1	Data.....	12
4.1.2	Informace.....	13
4.1.3	Znalosti.....	14
4.2	Prostředí WWW	16
4.3	Informační obsah v prostředí www	17
4.4	Webové technologie	17
4.4.1	Technologie client-side.....	18
4.4.2	Technologie server-side.....	33
4.4.3	Protokol HTTP.....	36
4.4.4	AJAX.....	37
4.4.5	Moderní trendy a přístupy k vývoji aplikací.....	38
4.5	Sémantika v prostředí WWW	40
4.5.1	Sémantika a HTML.....	40
4.5.2	Metadatový popis.....	41
4.6	Shrnutí	44

5	Metody a nástroje pro sběr a zpracování dat v prostředí WWW	46
5.1	Správa informačního obsahu v prostředí WWW.....	46
5.2	Web Content Management System	47
5.3	Redakční systémy a správa obsahu.....	49
5.3.1	Drupal	49
5.3.2	Joomla	51
5.3.3	WordPress	53
5.4	Shrnutí	55
5.4.1	Vkládání a aktualizace obsahu	56
5.4.2	Ukládání obsahu	56
5.4.3	Prezentace, sémantika a archivace	56
6	Metodika Web Information Content Management (WICM).....	58
6.1	Hypotéza dizertační práce	58
6.2	Cíle a požadavky navrhované metodiky	58
6.2.1	Zaměření metodiky	59
6.2.2	Identifikace výchozích požadavků a nedostatků současného stavu	59
6.3	Návrh metodiky	62
6.3.1	Výchozí prototyp	62
6.3.2	Výchozí formulace metodiky	65
6.3.3	Vkládání a aktualizace obsahu	66
6.3.4	Ukládání obsahu	68
6.3.5	Prezentace, sémantika a archivace	73
6.4	Formulace metodiky WICM	79
6.4.1	Konvence	79
6.4.2	Vkládání (tvorba) a aktualizace obsahu	80
6.4.3	Ukládání obsahu	81
6.4.4	Prezentace, sémantika a archivace	86
7	Ověření, zhodnocení přínosů a možností metodiky	90
7.1	Ověření pomocí metody Focus Groups	90
7.2	Řízené rozhovory s odborníky s praxí.....	92
7.3	Zhodnocení možných přínosů metodiky WICM	95
7.3.1	Přínosy pro autory obsahu	95

7.3.2	Přínosy pro vývojáře.....	95
7.3.3	Přínosy pro čtenáře obsahu	96
7.3.4	Přínosy v oblasti sémantiky	96
7.3.5	Ekonomické přínosy	96
7.4	Možnosti a doporučení pro další výzkum	97
8	Závěry dizertační práce	99
9	Seznam použitých zdrojů.....	105
10	Přílohy	114
10.1	Seznam obrázků.....	114
10.2	Seznam tabulek	115
10.3	Seznam zdrojových kódů.....	116
10.4	Seznam diagramů	116
10.5	Exportní formát JSON Schema	117
10.6	Publikační činnost autora	125

1 Úvod

„When something is such a creative medium as the web, the limits to it are our imagination.“

Timothy Berners-Lee

Internet a prostředí World Wide Web se za posledních více jak dvacet let rozšířilo téměř ke všem vrstvám obyvatel. Webových stránek je čím dál více. V současné době se jejich počet odhaduje na více než jednu miliardu a tento počet stále stoupá. Když v roce 1989 vytvořil Timothy Berners-Lee World Wide Web, byl to nejprve jen návrh systému pro správu vědeckých dokumentů v laboratořích CERN. Velmi záhy se však rozšířil do celého světa a stal se všeobecně používanou platformou. Přístup lidstva k datům a informacím se od té doby významně změnil.

V oblasti webových technologií je v současné době vidět řada změn v přístupech k vývoji, ukládání dat, ale i celkově architektuře a principům aplikací. HTML již není jen značkovací jazyk, ale obsahuje i popis různých aplikačních rozhraní (API) a napojení na další nástroje. Jazyk JavaScript získává stále více na popularitě. Jeho využití i na straně serveru je stále rozšířenější. Zároveň se setkáváme s trendem přesunu velké části funkcionality aplikací na stranu klienta. Tradiční pojetí webových aplikací, kde se při každém http požadavku načítala celá stránka se pomalu mění. Stále vyšší požadavky na dynamiku, rychlost a menší datovou náročnost tak daly vzniknout konceptu single-page aplikací. Server pak funguje primárně jako poskytovatel dat. Tento posun je urychlován i rozvojem mobilních zařízení. Serverová část aplikace pak může být společná pro webové i nativní mobilní aplikace. Tradiční relační databázové systémy jsou často nahrazovány NoSQL databázemi. Vznikají i služby, které kompletně nahrazují vlastní server a serverovou část aplikace. Typickým příkladem je například Google Firebase nebo Firestore.

Pro tištěná média, jako jsou noviny, časopisy a knihy, musí obsah před vydáním upravit odborní pracovníci ve specializovaných programech. Zpočátku i publikování obsahu na webu nebylo snadné a vyžadovalo znalost minimálně jazyka HTML. Postupně, zejména s rozvojem webových technologií a webových aplikací, se tato

možnost rozšiřuje i mezi širokou veřejnost. Dnes je publikování obsahu online snadnější, zejména díky redakčním systémům a jejich nástrojům. Roste tak počet mediálních serverů a portálů. Zároveň samotné vytváření a vydávání obsahu není doménou pouze odborníků a specializovaných pracovišť.

Lidé se dnes odklánějí od klasických tištěných médií, noviny a časopisy jsou nahrazovány webovými portály. K tomu dopomáhá i rozvoj mobilních zařízení jako jsou chytré telefony, tablety nebo elektronické čtečky. Přístup uživatelů na internet z těchto zařízení neustále roste a v současnosti se odhaduje na zhruba 50% všech přístupů.

Internet a web dnes mají veliký vliv na lidskou společnost a mění řadu lidských činností. To platí především pro oblast publikování informací. Řada médií prostřednictvím webu zpočátku jen poskytovala obsah v digitální podobě. Prostředí World Wide Web však nabízí oproti tištěnému formátu celou řadu výhod. Dynamika a možnosti interakce umožňují dříve nevídané zapojení čtenářů. Nejde však jen o možnost komentování, ale i zapojení do samotného procesu žurnalistiky. Před několika lety se objevil např. koncept tzv. otevřené žurnalistiky, jehož velkým propagátorem byl anglický list The Guardian. Jeho hlavním principem je právě zapojení čtenářů do samotného procesu vzniku článků a myšlenka, že autor není jedinou a pravdivou autoritou. Jedním z inovativních a viditelných znaků bylo především zveřejňování publikačního plánu redakcí tak, aby se i veřejnost mohla zapojit a přinést další informace. Dnes se tento koncept postupně vyvíjí a stále se hledají možnosti využití moderních technologií. V českých médiích a veřejném prostoru se spíše používá pojem občanská žurnalistika. Nejviditelnějším příkladem je pravděpodobně aplikace iReportér České televize.

Jedním z limitujících faktorů jsou v současnosti nástroje a metody pro práci s obsahem. Velká vydavatelství musí investovat prostředky do vlastních aplikací a systémů. V prostředí open source a zejména redakčních systémů je v této oblasti zaznamenáván jen velmi pomalý posun. Standardizace a sjednocení některých postupů by mohla být významným přínosem.

2 Cíle práce

2.1 Hlavní cíl

Hlavním cílem dizertační práce „Metody sběru a zpracování dat v prostředí WWW“ je navrhnout metodiku pro vkládání, aktualizaci, uložení a prezentaci informačního obsahu v prostředí WWW (World Wide Web). Důraz bude kladen i na archivaci, sdílení a možnosti transformace obsahu pro využití v různých zařízeních (např. mobilní telefony, tablety apod.). Nedílnou součástí práce je nalézt a ověřit vhodné metody a nástroje, které budou tuto metodiku tvořit. V dalším textu bude název metodiky uváděn pod zkratkou WICM (Web Information Content Management).

2.2 Dílčí cíle dizertační práce

1. Prozkoumat současný stav poznání v oblasti „data, informace, znalosti“ a vymezit pojem „informační obsah“ v prostředí WWW.
2. Prozkoumat současný stav a rozvoj webových technologií vzhledem k zaměření hlavního cíle.
3. Prozkoumat současný stav a rozvoj technologií a nástrojů pro sémantiku v prostředí WWW.
4. Analyzovat používané metody a nástroje pro sběr a zpracování dat v prostředí WWW vzhledem k zaměření hlavního cíle práce.
5. Po splnění hlavního cíle ověřit navrženou metodiku.
6. Zhodnotit možné přínosy navržené metodiky.
7. Synteticky zhodnotit dosažené výsledky a formulovat závěry a doporučení pro další výzkum.

3 Metodický postup

Předkládaná práce je tematicky zaměřena na prostředí a technologie pro World Wide Web.

V první části práce je provedeno vymezení nezbytných pojmů a literární přehled. Metodicky je tato část založena na studiu a analýze informačních zdrojů české i zahraniční odborné a vědecké literatury, časopisů a sborníků z konferencí. Zřetel je brán zejména na vědecké publikace indexované v citačních databázích Scopus a Web of Science. Díky rychlému rozvoji v oblasti informačních a komunikačních technologií je využito také respektovaných odborných webových stránek a portálů. Nedílnou součástí studia a analýzy je také komparace vybraných informačních zdrojů.

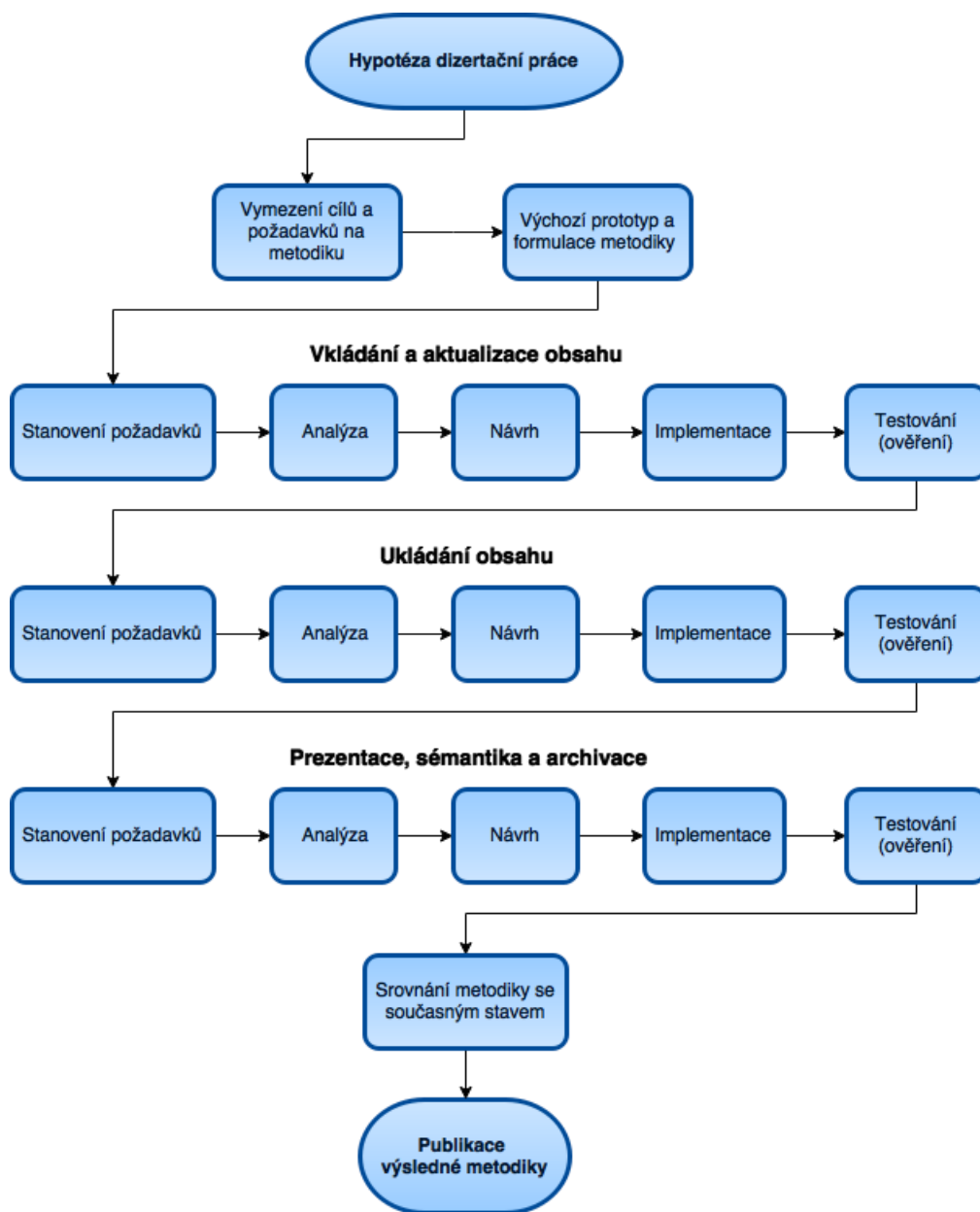
Dále je proveden přehled současného stavu používaných technologií a nástrojů. V této oblasti jsou analyzovány nejvýznamnější redakční systémy z hlediska vkládání, aktualizace, uložení a prezentace informačního obsahu.

Praktická část dizertační práce popisuje postup tvorby metodiky WICM. Syntézou poznatků z předchozích kapitol je nejprve stanovena hypotéza dizertační práce a následně jsou definovány výchozí cíle a požadavky na navrhovanou metodiku.

Postup tvorby metodiky WICM vychází z principů prototypového přístupu (strategie) pro vývoj informačních systémů. Nejprve je vytvořen výchozí (zjednodušený) prototyp softwarového modulu pro vkládání, aktualizaci, ukládání a prezentaci obsahu. Prototyp je pak opakovaně podroben fázím Stanovení požadavků, Analýzy, Návrhu, Implementace a Testování. V rámci cyklů je prototyp postupně upravován tak, aby vyhovoval nově identifikovaným požadavkům a nahradil či opravil identifikované nedostatky. Každý průchod cyklem je zaměřen primárně (ne však výlučně) na jednu oblast řešení. Tyto oblasti jsou definovány v kapitole 6.2. Během jednotlivých cyklů vývoje je postupně navrhována metodika WICM a pomocí prototypu i ověřována. Celkový postup tvorby metodiky je znázorněn na obrázku č.1.

Výsledná metodika je formulována obdobně jako současné webové standardy a specifikace. Její součástí je sada doporučení a pravidel, doménový model ve formě diagramu tříd UML a definice formátu pro výměnu dat pomocí notace JSON Schema.

V posledních částech práce je provedeno ověření finální verze metodiky s pomocí metody Focus Groups a série polo-strukturovaných rozhovorů. Následně jsou zhodnoceny možné přínosy navržené metodiky. V poslední kapitole práce jsou shrnuty všechny poznatky a formulovány závěry a doporučení pro další výzkum.



Obrázek č. 1 - Postup tvorby metodiky; zdroj: vlastní zpracování

3.1 Metodický aparát

Pro zpracování dizertační práce bylo využito základních vědeckých metod komparace, analogie, analýzy a syntézy poznatků. V následujících podkapitolách jsou popsány další pojmy a metody v kontextu jejich využití v disertační práci.

3.1.1 Metodika

Pojem metodika je v různé literatuře většinou definován velmi podobně. Autoři se shodují na tom, že jde o postup, nebo návod na řešení určitého problému, který vymezuje jednotlivé kroky a použité metody. Pro zaměření předkládané práce je nejbližší definice dle Molnára (2012): *Metodika představuje souhrn doporučených praktik a postupů pro vykonání nějaké činnosti, například nauka o metodách vědecké práce. Ve vývoji software metodika představuje souhrn doporučených praktik a postupů, pokrývajících celý životní cyklus vytvářené aplikace. Pro řešení dílčích problémů mohou být v rámci nasazení metodiky uplatněny specifické postupy – metody.*

3.1.2 Metoda

Slovo metoda pochází z řečtiny (*meta hodos – cesta někam*). Metoda je způsob (postup, návod), jak dosáhnout nějakého cíle. Důležitým aspektem je, že na takto dosažený cíl nebo poznatek se ostatní mohou spolehnout. Poznání dosažené určitou metodou tak může být označováno za pravdivé. (Hendl, 2008; Descartes, 1992)

3.1.3 Prototypový přístup

Prototypový přístup popisuje v literatuře prvně Bally et al. (1977) jako strategii pro vývoj informačních systémů: *„In the prototype strategy, an initial and usually highly simplified prototype version of entire system is designed, implemented, tested and brought into operation. Based on the experience gained in the operation of the first prototype, a revised requirement is established, and a second prototype designed and implemented. The cycle is repeated as often as is necessary to achieve a satisfactory operational system...“*

Prototyp lze v modernějším pohledu chápat jako neúplný software, funkční část, nebo zjednodušenou implementaci systému (Arnowitz et al., 2006). Hlavním principem prototypového přístupu je tedy postupná tvorba prototypů a jejich evaluace. Tento přístup byl vyvinut jako doplněk k tradičním metodám návrhu a vývoje software (Weinberg, 1991). Mezi hlavní výhody dle Tvrdíkové (2001) patří:

- celistvost řešení problému,
- zajištění účasti uživatelů na vývoji informačního systému,
- výběr nejlepšího návrhu z více návrhů,
- oddělení fáze návrhu od fáze implementace a tím snížení nákladů na odstraňování dříve neodhalených nedostatků informačního systému,
- možnost odzkoušení fungování systému v podmínkách reálného světa,
- zkrácení doby od specifikace požadavků naIS/IT do zavedení systému do užívání,
- účast netechnicky orientovaných manažerů na řešení informačního systému.

3.1.4 Metoda Focus Groups

Metoda Focus Groups je jednou ze základních metod, která se používá při výzkumu v oblasti vývoje systémů, použitelnosti, User experience, ale i marketingu. Jedná se o neformální metodu, při které se zjišťují potřeby uživatelů ještě před započítím designových prací (Nielsen, 1997). Jde o formu moderované diskuze, při které účastníci hovoří s moderátorem o svých potřebách, pocitech, přáních, ale i reakcích na koncepty (U.S. Dept. of Health and Human Services, 2013).

3.1.5 UML a doménový model

UML je grafický jazyk pro vizualizaci, specifikaci, konstrukci a dokumentaci artefaktů v distribuovaných objektových systémech (OMG, 2015a). Jde tedy především o soubor notací, který se používá při návrhu software a nemá nic společného s procesem vývoje (Barclay a Savage, 2003). Cílem UML je pak poskytnout systémovým architektům, softwarovým inženýrům a vývojářům nástroje pro analýzu, návrh a implementaci softwarových systémů a business procesů (OMG, 2015b).

Při navrhování dat a datových struktur je zpravidla prvním krokem vytvoření doménového modelu, typicky ve formě diagramu tříd v UML (Cadle et al., 2014). Doménový model oproti diagramu tříd představuje spíše náčrt systému. Neobsahuje metody, má pouze důležité atributy, atributy nemají datové typy a je platformě nezávislý (Čápka, 2013).

4 Literární přehled

V následujících kapitolách jsou nejprve vymezeny základní pojmy data, informace a znalosti vzhledem k zaměření hlavního cíle práce. Na základě definice prostředí World Wide Web (WWW) a těchto pojmů je pak vymezen pojem informační obsah. Další kapitoly se pak zabývají důležitými technologiemi prostředí WWW a způsoby jejich užití, které jsou nezbytné pro splnění hlavního cíle.

4.1 Data, informace, znalosti

4.1.1 Data

Definice dat se napříč odbornou literaturou celkem shoduje a definice jsou si podobné. Například dle Davenporta a Prusaka (2000) jsou data objektivní fakta o událostech. Většina definic se pak shoduje na tom, že data mohou být např. čísla, písmena, symboly, obrazy, zvuky apod.

Podrobněji lze data charakterizovat následovně (Vymětal et al., 2005):

- *Data představují obvykle něco, co se dá v praxi získat experimentem, měřením, pozorováním nebo šetřením.*
- *Data tedy objektivně zobrazují stavy nebo vlastnosti objektů nebo probíhající procesy v reálném prostředí kolem nás, a to nezávisle na našem vědomí.*
- *Základem dat jsou znaky, což mohou být zejména číslice, písmena a symboly (tečky, čárky, noty apod.).*
- *V širším pojetí se pojmu data používá pro numerické (dnes především), textové, obrazové a zvukové údaje, aniž posuzujeme, co znamenají pro příjemce. Jde nám především o jejich vyjádření a uložení pro perspektivní zpracování.*
- *Data můžeme považovat za jednoduché reprezentační nástroje faktů s jednorozměrným, jedinečným významem.*

Podle ČSN ISO/IEC 2382-1:1998¹ je pojem data definován jako: „*Opakovaně interpretovatelná formalizovaná podoba informace vhodná pro komunikaci, vyhodnocování nebo zpracování.*“ Tuto definici přebírá i nejnovější ISO/IEC 2382:2015². Tato norma ještě není do ČSN zapracována.

4.1.2 Informace

V případě definice pojmu informace je situace složitější. Má mnoho významů a definic tak existuje celá řada. V různých vědních oborech se k tomuto pojmu přistupuje odlišně. V následujícím textu jsou vybrané definice, které jsou relevantní k zaměření práce a oboru Informační management.

Za zakladatele definice informace je považován Norbert Wiener. Ten říká, že „*informace je název pro obsah toho, co se vymění s vnějším světem, když se mu přizpůsobujeme a působíme na něj svým přizpůsobováním.*“ (Cejpek, 2005).

Mezi jednodušší definice se řadí např. (Hlavenka, 1997): „*Data, která nesou význam pochopitelný lidmi.*“

Katuščák et al. (1998) definuje informaci takto: „*V nejobecnějším slova smyslu se informace chápe jako údaj o reálném prostředí, o jeho stavu a procesech v něm probíhajících. Informace snižuje nebo odstraňuje neurčitost systému (např. příjemce informace); množství informace je dáno rozdílem mezi stavem neurčitosti systému (entropie), kterou měl systém před přijetím informace a stavem neurčitosti, která se přijetím informace odstranila...*“ Pro oblast informatiky pak ještě přidává, že za „*informaci považuje kvantitativní vyjádření obsahu zprávy. Za jednotku informace se ve výpočetní technice považuje rozhodnutí mezi dvěma alternativami (0, 1) a vyjadřuje se jednotkou nazvanou bit.*“

¹ http://www.technicke-normy-csn.cz/369001-csn-iso-iec-2382-1_4_26691.html

² <https://www.iso.org/standard/63598.html>

Dle normy ISO/TR 9007:1987³ je definována informace takto: „Any kind of knowledge about things, facts, concepts, etc. of a universe of discourse that is exchangeable among users. Although exchangeable information necessarily will have a representation form to make it communicable, it is the interpretation of this representation (the meaning) that is relevant in the first place.“

„Data presented in readily comprehensible form to which meaning has been attributed within the context of its use. In a more dynamic sense, the message conveyed by the use of a medium of communication or expression. Whether a specific message is informative or not depends in part on the subjective perception of the person receiving it.“ (Reitz, 2017)

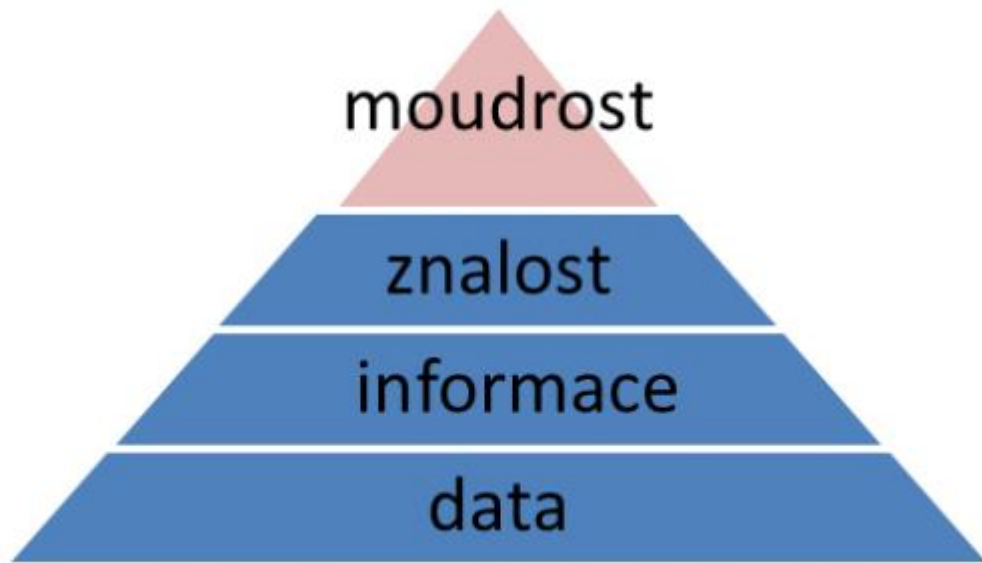
Podle české státní normy ČSN ISO/IEC 2382-1:1998 je pojem informace definován jako poznatek (znalost) týkající se jakýchkoliv objektů, např. faktů, událostí, věcí, procesů, myšlenek nebo pojmů, které mají v daném kontextu specifický význam. Dle nejnovější normy ISO/IEC 2382:2015 je definice: „knowledge which reduces or removes uncertainty about the occurrence of a specific event from a given set of possible events“.

Vaníček (2007) v knize Teoretické základy informatiky definuje pojem informace takto: „Informace je základním pojmem teorie poznání. Lze ji neformálně popsat jako veličinu, která určuje stupeň naší jistoty, že nastane nějaká událost. Tedy míru toho, do jaké jsme poznali svět kolem nás, případně jakýkoliv jiný reálný nebo předpokládaný systém.“

4.1.3 Znalosti

V literatuře se často mluví o hierarchii, nebo pyramidě data → informace → znalosti. Tento vztah znázorňuje lépe následující Obrázek č. 2.

³ <https://www.iso.org/standard/16549.html>



Obrázek č. 2 - Znalostní pyramida; zdroj: (Chytková a Černý, 2012)

Cejpek (2005) tyto tři pojmy vysvětluje následovně:

Data – základní jednotka informační vědy a informačního procesu. Měřitelná v jednotkách bajt a bit. Data zobrazují stavy nebo vlastnosti objektů. Základem dat jsou znaky.

Informace – data, kterým příjemce přisuzuje určitý význam na základě znalostí, zkušeností a vědomostí, kterými disponuje. Informace musí být pro příjemce nová.

Znalosti – vznikají zasazením určitého množství informací do kontextu. Jsou tedy definovány v kontextu jiných informací.

Na obrázku č. 2 je ještě znázorněn pojem moudrost. Ten se v informatice příliš neobjevuje. Vaníček (2007) ji dává do kontextu informací o znalostech: „*Informace nemusí být pouze jednoúrovňové, sdělující nějaká fakta o předmětech a jevech kolem nás. Mohou to být také informace o informacích, které zaznamenávají vlastnosti předmětů a jevů a vztahy mezi nimi. Takovým „informacím o informacích“ říkáme zpravidla znalost. I znalosti můžeme zaznamenat daty a data interpretovat jako znalosti. V dané souvislosti se někdy užívá i slova moudrost. Je to opět speciální znalost (tedy informace). Jde o znalost toho, jak informace a znalosti užívat v souladu s našimi zájmy a pravidly, které respektujeme (například pravidly etiky). Tedy další vyšší úroveň našeho poznání.*“

Další definice pojmu znalost se týkají jiných oborů a vzhledem k zaměření práce jsou irelevantní.

4.2 Prostředí WWW

Definici pojmu World Wide Web (WWW) nabízí konsorcium W3C (W3C, 2004) v doporučení Architecture of the World Wide Web: „*The World Wide Web (WWW, or simply Web) is an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI).*“

Autorem původního konceptu WWW je Timothy Berners-Lee. Ten v roce 1989 v laboratořích CERN (the European Laboratory for Particle Physics in Geneva) vytvořil návrh *Information Management: A Proposal*⁴, který obsahoval jazyk HTML, URL a protokol HTTP. Prvotní ideou bylo zpřístupnění a organizace vědeckých dokumentů v rámci CERN. Avšak místo toho, aby je uživatelé stahovali do svých počítačů, měli být dokumenty propojeny navzájem odkazy (Raggett, 1998). To je dodnes klíčová vlastnost jazyka HTML a WWW obecně.

Základním principem je využití síťové architektury klient-server a celosvětové sítě internet. Ta spojuje dva druhy počítačů: servery, ty poskytují dokumenty a klienty, kteří je přijímají a zobrazují. To, co se odehrává na serveru je z hlediska technologií označováno jako *server side*, zatímco aktivity na klientských počítačích jsou *client side*. (Musciano a Kennedy, 2007)

Jako klient je v literatuře často označován pouze webový prohlížeč. Ten komunikuje prostřednictvím protokolu HTTP se serverem. Jako *webový server* je pak označována aplikace, která zpracovává HTTP požadavky, na jejich základě poskytuje webový obsah.

⁴ (<http://www.w3.org/History/1989/proposal.html>)

4.3 Informační obsah v prostředí www

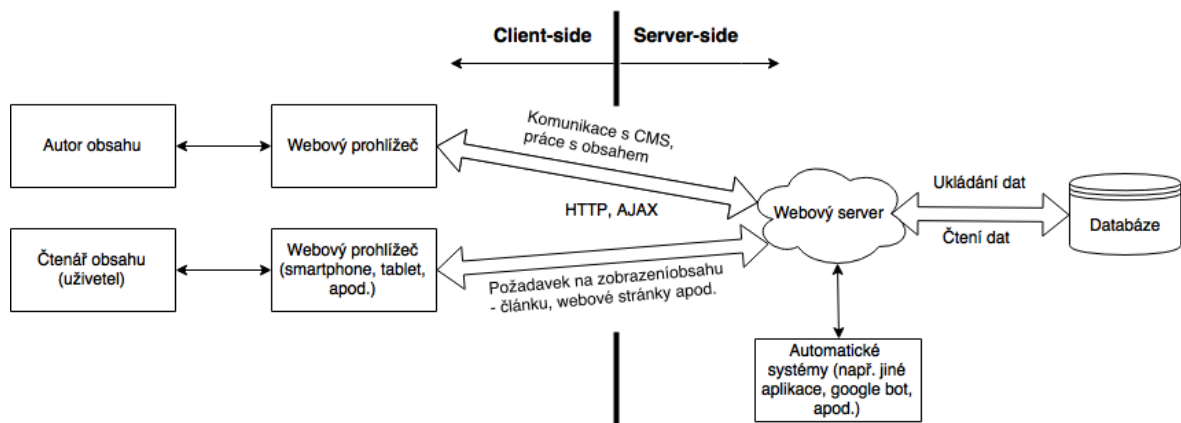
Pojem informační obsah je definován pouze v teorii informace. Anglický překlad pak zní *information value*. Přesnější překlad pojmu informační obsah, tak jak je užíván v této práci, je *information content*. V prostředí WWW nebyl dosud termín *informační obsah* exaktně definován, i když většina veřejnosti chápe jeho význam. Na základě syntézy poznatků a definic z předchozí kapitoly je možné jej pro potřeby práce blíže vymežit.

Informační obsah je zpráva, která je předávána prostřednictvím internetového prohlížeče uživateli. Tato zpráva je sestavena z dat uložených na straně serveru. Tato data již byla setříděna a kategorizována a k uživateli se dostávají v určité formě. Na základě některých definic by se pak tento obsah mohl nazývat také znalostí (*informace o informacích* – Vaníček, 2007). Pro potřeby této práce však bude užito pojmu informační obsah, který je pochopitelnější pro širokou veřejnost. Význam slova *informační* tak vychází spíše z definice Slovníku spisovného jazyka českého.

Z praktického hlediska je informační obsah to, co si uživatel zobrazí v rámci webové stránky. Nejčastěji se tedy jedná o článek, webovou stránku, fotogalerii apod. Pokud to vztáhneme k modernímu značkovacímu jazyku HTML5 a jeho sémantice, je informační obsah především to, co je uvnitř elementu `<article>`. Práce se tak v dalších kapitolách bude zabývat obsahem, který je na webových stránkách v tomto elementu prezentován.

4.4 Webové technologie

Webové technologie se obecně dělí podle toho, zda se týkají serverové části (*server side*), nebo klientské (*client side*). Toto rozdělení s ohledem na zaměření práce ilustruje obrázek č. 3. Vybrané technologie s významem pro cíle dizertační práce budou popsány v následujících dvou kapitolách.



Obrázek č. 3 - Webové technologie - rozdělení client side, server side; zdroj: vlastní zpracování

4.4.1 Technologie client-side

HTML

Hypertextový značkovací jazyk HTML vznikl v roce 1989 v laboratořích CERN a jeho autorem je Tim Berners-Lee. Návrh jazyka byl založen na standardu SGML (ISO 8879:1986 Information processing – Text and office systems - Standard Generalized Markup Language)⁵ a obsahoval jen několik elementů.

Zásadní inovací byl tzv. **Hypertext** – propojení dokumentů odkazy. Raggett (1998) o tom přesněji píše: „*The SGML elements used in Tim's HTML included; H1 through H6; OL; UL; LI and various others. What SGML does not include, of course, are hypertext links: the idea of using the anchor element with the HREF attribute was purely Tim's invention, as was the now-famous 'www.name.name' format for addressing machines on the Web.*“

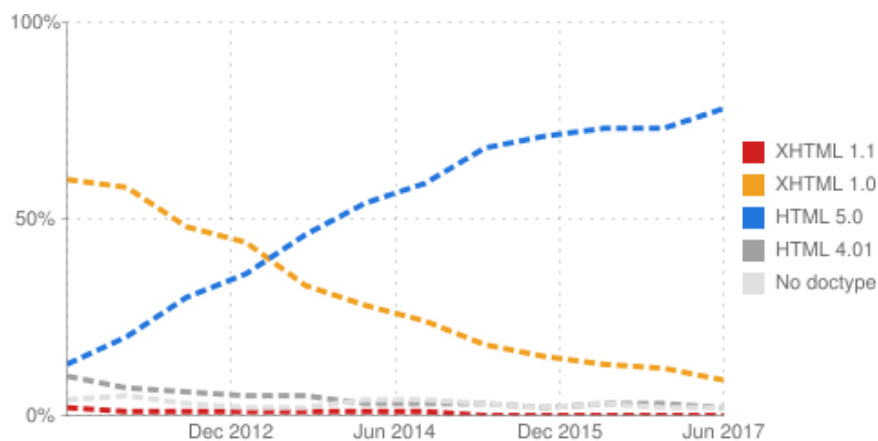
Původní verze jazyka HTML nebyla nikde standardizována a verze 1.0 neexistuje (Keith, 2010). První oficiální specifikace tak nese označení HTML 2. Tu vytvořila v roce 1994 pracovní skupina *HTML working group* sestavená v rámci organizace The Internet Engineering Task Force (IETF) (Raggett, 1998).

⁵ <https://www.iso.org/standard/16387.html>

Koncem roku 1994 pak byla založena organizace World Wide Web Consortium (W3C), v jejímž čele stál Timothy Berners-Lee (Raggett, 1998; W3C, 2015a). Ta převzala vývoj jazyka HTML a souvisejících webových technologií. První specifikací HTML pod W3C tak byla verze 3.2 z roku 1997 (Raggett, 1997). Stručný přehled vývoje jednotlivých verzí je uveden v následující tabulce.

1989	Původní HTML
1994	HTML 2
1997	HTML 3.2
1999	HTML 4.01
2000	XHTML 1.0
2001	XHTML 1.1
Vývoj přerušen	XHTML 2
2014	HTML5
2016	HTML 5.1
2017	HTML 5.2

Tabulka č. 1 - Historie verzí HTML; zdroj: vlastní zpracování



Obrázek č. 4 - Rozložení verzí HTML na webových stránkách; zdroj: (PowerMapper, 2017)

HTML 4.01

První široce rozšířenou specifikací jazyka HTML byla verze 4.01⁶. Ta byla vydána jako W3C Recommendation 24. prosince 1999 jako revize HTML 4.0 (W3C, 1999). Dodnes je možné najít webové stránky využívající tento standard. Rozložení jednotlivých verzí HTML používaných na webových stránkách ukazuje Obrázek č. 4.

Z hlediska zaměření práce je důležitá většina specifikace. Důležitým aspektem je syntaxe jazyka. Ta je založena na metajazyku SGML a je tak jeho aplikací již od verze 2.0 (Berners-Lee a Connolly, 1995; W3C, 1999). Nejdůležitější prvky jazyka je možné na základě specifikace definovat následovně:

- Jména elementů jsou vždy case-insensitive (nezáleží na velikosti písmen).
- Některé elementy nemusí mít uzavírací značku.
- Jména atributů jsou vždy case-insensitive (nezáleží na velikosti písmen).
- Hodnoty atributů nemusí být vždy uzavřené v uvozovkách (jednoduchých, nebo dvojitých).
- Některé atributy mohou být bez hodnoty.

XHTML

V roce 2000 byla vydána specifikace XHTML. Ta byla v roce 2002 revidována (W3C, 2002). Dodnes je používána na řadě webových stránek. Tato specifikace je přepracováním specifikace HTML 4.01 s použitím pravidel XML. Nepřináší téměř nic nového. Vypouští např. atribut *name*, jako identifikátor pro elementy *a*, *applet*, *form*, *frame*, *iframe*, *img*, a *map*. Místo něj se používá pouze atribut *id*.

Rozdíly oproti HTML 4.01 vychází z aplikace XML namísto SGML a jsou především v použitelné syntaxi. Obecně musí být splněna pravidla syntaxe XML.

V roce 2001 se pak ještě objevila specifikace XHTML 1.1⁷, která je abstraktní modularizací XHTML 1.0 (W3C, 2010). Hlavním přínosem byla zmíněná modulární architektura. Specifikace se tak skládá z řady modulů, které řeší určité oblasti.

⁶ <https://www.w3.org/TR/1999/REC-html401-19991224/>

⁷ <http://www.w3.org/TR/xhtml11/>

Následoval vývoj specifikace XHTML 2, ale ve vývoji byla i specifikace HTML 5 pod skupinou WHATWG. Tu později převzalo i W3C na úkor HTML 2. První veřejný *working draft* byl vydán v roce 2008 pod označením HTML5. Standard XHTML 1.1 pak již nedosáhl tak širokého využití jako XHTML 1.0.

Standard 1.1 vychází z varianty XHTML 1.0 Strict. Zaměříme-li se na rozdíly mezi těmito specifikacemi, z hlediska informačního obsahu se jedná pouze o úplné vypuštění atributu `name` (u elementů *a* a *map*) a přidání kolekce elementů "ruby" (W3C, 2010).

HTML5

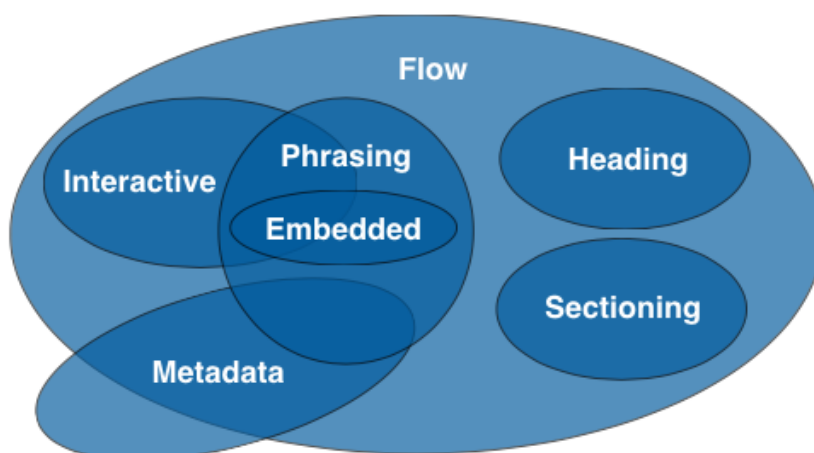
Specifikace HTML5 se na webových stránkách využívá již delší dobu. První verze specifikace ve fázi „recommendation“ byla vydána v roce 2014 (W3C, 2014b), ale využívat se na webových stránkách začala dříve. Dnes je tato specifikace nejrozšířenější používanou verzí – viz. Obrázek č. 4.

HTML5 zahrnuje specifikaci abstraktního jazyka a zároveň různé sady API (Goldstein et al., 2011). Z hlediska syntaxe umožňuje používat dva typy – „HTML 5“ (MIME type *text/html*) a „XHTML 5“ založenou na XML/XHTML (MIME type *text/xml*, *application/xml*, *application/xhtml+xml*) (W3C, 2017b). Syntaxe HTML 5 je volnější a pro vývojáře doporučená. Vychází z původní verze HTML 4.01, ale již se nejedná o aplikaci XML ani SGML a je pevně definovaná ve specifikaci jazyka. Nově je např. nutné definovat kódování, změnila se notace DOCTYPE nebo je možné přímé užití SVG a MathML.

Elementy

HTML5 specifikuje kategorie, do kterých může každý element patřit. Každý element může spadat do více kategorií, ale také do žádné. Tyto kategorie jsou používány především při definování tzv. *content modelu*. Ten definuje, co může být obsahem každého elementu. Jinými slovy, které elementy, mohou být uvnitř jako přímý potomek. Základní kategorie jsou z hlediska svých charakteristik *Metadata content*, *Flow content*, *Sectioning content*, *Heading content*, *Phrasing content*, *Embedded content*,

Interactive content (W3C, 2017b). Nejvíce elementů je typu *Flow content*. Rozložení a prolnutí jednotlivých kategorií ukazuje obrázek č.4.



Obrázek č. 5 - Typy elementů HTML5; zdroj: (W3C, 2017b)

Budoucí vývoj jazyka HTML

Důležité pro informační obsah a hlavní cíl práce je to, že webové stránky a portály bývají provozovány po delší dobu a často mění vzhled, redakční systém apod. Jsou tedy často přepracovávány. To s sebou může nést i změnu použité specifikace HTML. Aktuální i budoucí specifikace HTML tak počítají se zpětnou kompatibilitou prvků, které byly v novějších verzích odebrány. Tyto prvky jsou označovány pouze jako „obsolete“ a ne „deprecated“. Jsou stále součástí specifikace, pro usnadnění přechodu k novějším verzím. Validátory by je pak měly označit varováním (W3C, 2017a). Přesněji se o tom zmiňuje Keith (2010): *„Because HTML5 aims to be backwards compatible with existing content, the specification must acknowledge previously existing elements even when those elements are no longer in HTML5. This leads to a slightly confusing situation where the specification simultaneously says, “authors, don’t use this element” and, “browsers, here’s how you should render this element.” If the element were deprecated, it wouldn’t be mentioned in the specification at all; but because the element*

is obsolete, it is included for the benefit of browsers.“ Tento přístup následuje i zatím poslední verze specifikace HTML5 – HTML 5.2⁸, vydaná v prosinci 2017.

Vedle organizace W3C však existuje ještě pracovní skupina WHATWG (The Web Hypertext Application Technology Working Group). Ta je autorem původních návrhů Web Applications 1.0, Web Forms 2.0 a HTML 5, které byly později převzaty W3C jako základ HTML5. Celá skupina vznikla ve stejné době, kdy W3C pracovalo na XHTML 2 a je tvořena spíše zástupci výrobců prohlížečů. V současné době vyvíjí specifikaci HTML Living Standard (HTML LS) (WHATWG, 2018). Existují tak dvě organizace, které se věnují webovým standardům. Vztah mezi W3C a WHATWG a jejich dvěma standardy – HTML5 a HTML LS není úplně jasný a jednoduchý. Někteří členové WHATWG spolupracovali i na specifikaci HTML5. Dle Shankland (2014) je HTML5 obraz spíše toho, co je implementováno ve webových prohlížečích a HTML LS spíše toho, co by mohlo být. Logické však je, že poslední slovo mají právě výrobci webových prohlížečů. Ti totiž implementují funkce a v závěru určují, co a jak je možné používat.

CSS

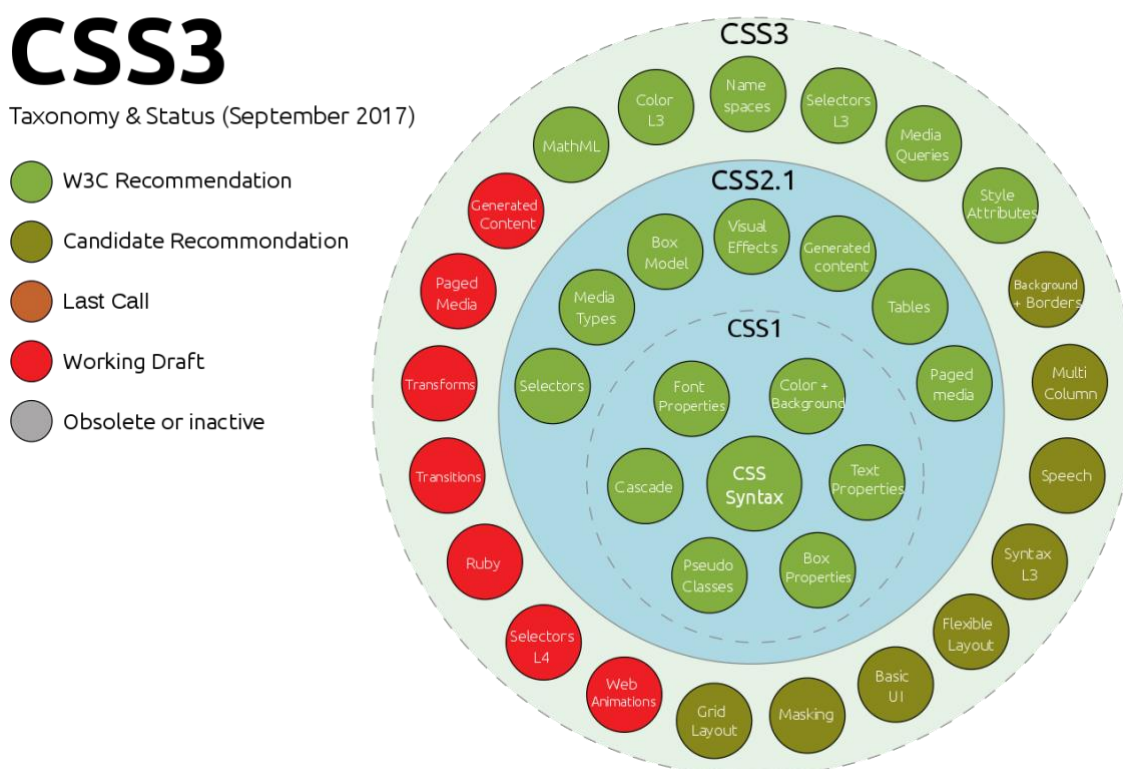
CSS je zkratka pro Cascading Style Sheets – v překladu šablony kaskádových stylů. Jedná se o značkovací jazyk, kterým je možné určovat vzhled HTML dokumentů a celkově webových stránek.

Jazyk HTML a web byl původně navržen pouze pro textové dokumenty (Raggett, 1998; Schafer, 2009). Ve verzi 3.2 byly přidány některé formátovací prvky, což vedlo k obtížím pro vývojáře. Pro vyřešení problémů, s tím spojených, tak vzniklo CSS. Od verze 4.0 by tak veškeré formátování mělo být odděleno v kaskádových stylech, nezávisle na obsahu (W3Schools, 2016).

V současné době jsou ve vývoji jednotlivé moduly CSS. Ačkoli se často mluví o CSS3, dle Gasstona (2014) žádné CSS3 není. Moderní CSS má modulární architekturu a namísto jedné velké specifikace jsou spravovány odděleně jednotlivé moduly. Je tedy

⁸ <https://www.w3.org/blog/2017/12/html-5-2-is-done-html-5-3-is-coming/>

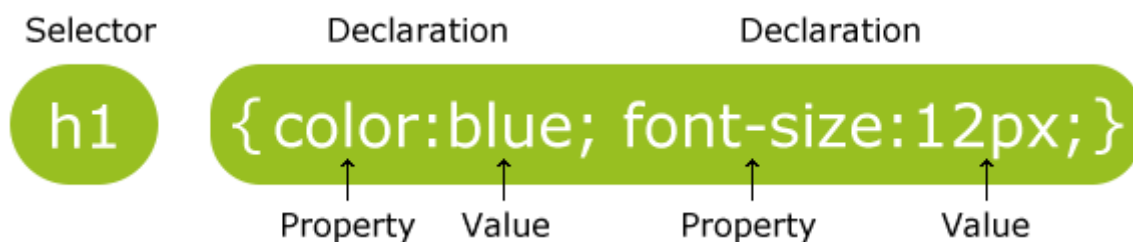
neustále ve vývoji. Přesněji se o tom Gasston (2014) zmiňuje následovně: „CSS3 is a specification in flux. Some parts of the spec are considered stable and have been well implemented in modern browsers; other parts should be considered experimental and have been partially implemented to varying degrees; yet others are still theoretical proposals and have not been implemented at all. Some browsers have created their own CSS properties that aren't described in any CSS3 specification and perhaps never will be.“ Jako v případě HTML5 je pak nejdůležitější to, co je implementováno ve webových prohlížečích. Schéma modulů CSS3 pak ukazuje následující Obrázek č. 6:



Obrázek č. 6 - Moduly CSS; zdroj: (Wikimedia Commons, 2017)

CSS je stylovací jazyk, který se používá jako doplněk k HTML dokumentům. Syntaxe jazyka je velmi jednoduchá. Je tvořena především třemi základními prvky – selektor, vlastnost, hodnota. Dále jsou k dispozici komentáře. Ty se zapisují obdobně jako ve většině programovacích jazyků do sekvence `/* ... */`. Další prvky pak přináší jednotlivé

moduly, tak jak jsou postupně ve vývoji. Mezi nejdůležitější patří např. modul Media Queries⁹ s pravidlem @media.



Obrázek č. 7 - Syntaxe jazyka CSS; zdroj: (W3Schools, 2016)

Kaskádové styly je pak možné připojit k HTML dokumentům několika způsoby (Keith, 2010; Gasston, 2014; W3Schools, 2016):

- Externě v souboru, nebo souborech s příponou *.css, připojené v sekci <head>.
 - <link rel="stylesheet" type="text/css" href="mystyle.css">
- Interně v elementu <style> v sekci <head>.
- Interně tzv. *scoped* elementem <link> kdekoli v sekci <body>.
 - <style scoped>
- Inline zápis v atributu style.
 - <h1 style="color:blue;margin-left:30px;">
- Pomocí direktivy @import uvnitř samotných stylů.

Pokud je použita kombinace zápisů pak platí, že čím blíže je předpis cílovému elementu, tím má větší prioritu, dle (Gasston, 2014) platí následující pořadí (1 - nejnižší; 3 - nejvyšší):

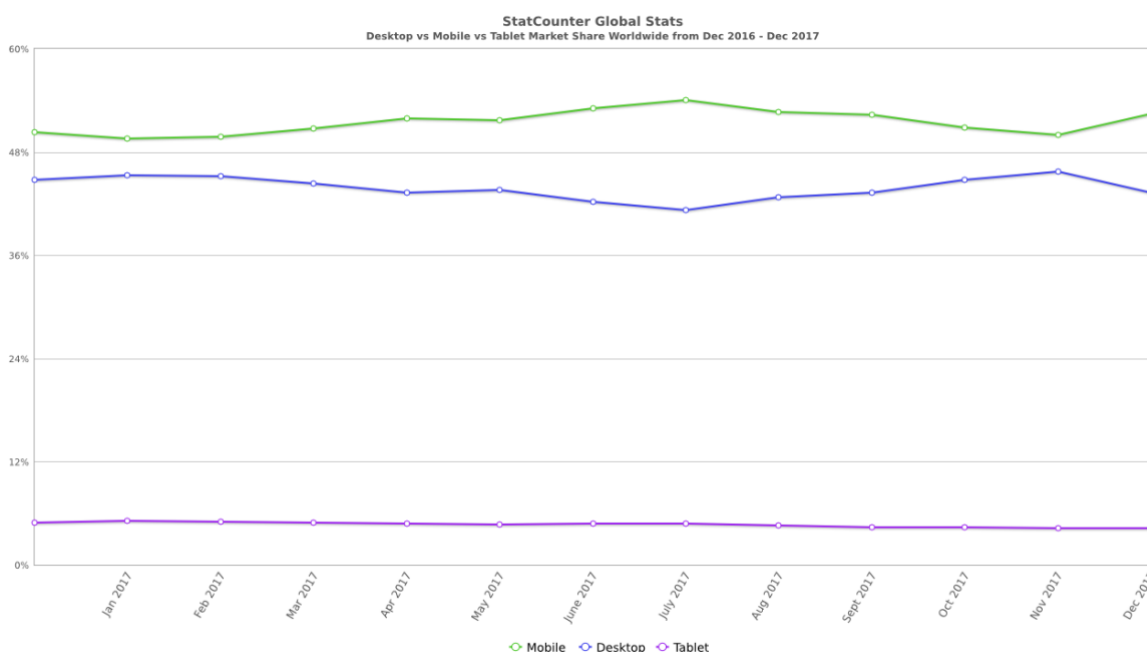
1. Výchozí styly prohlížeče.
2. Externí a interní styly (v sekci <head>).
3. Inline styly (uvnitř HTML elementů).

⁹ <https://www.w3.org/TR/css3-mediaqueries/>

Obrovskou výhodou pro informační obsah je pak to, že při změně celkového vzhledu webových stránek jej není nutné nijak upravovat. Stačí připojit jinou šablonu stylů.

Media Queries a rezpozivita webových stránek

Velmi důležitou vlastností webových stránek je jejich schopnost přizpůsobit zobrazení různým velikostem výstupních zařízení. Dle různých statistik je v současné době celosvětově realizováno více než 50% přístupů z mobilních zařízení – viz. Obrázek č. 8. Důležitost optimalizace webových stránek dokládá např. fakt, že nejrozšířenější vyhledávač Google penalizuje nepřizpůsobené webové stránky při hledání z mobilních zařízení (Google, 2015).



Obrázek č. 8 - Statistika přístupů k webovým stránkám dle typu zařízení; zdroj: (Statcounter, 2017)

K optimalizaci webových stránek pro zobrazení na různých zařízeních se využívá CSS modul Media Queries. Díky němu je možné zobrazovat různé styly v závislosti na typu média, na kterém je webová stránka zobrazována (W3C, 2012). Obsah se tedy může přizpůsobit dispozicím obrazovky, na které je zobrazen. Toto chování ilustruje Obrázek č. 9.



Obrázek č. 9 - Princip responzivního designu; zdroj: (WebAppRater, 2017)

Moderní přístupy k tvorbě CSS

Při tvorbě rozsáhlých webových stránek a portálů mají často zdrojové kódy kaskádových stylů rozsah v řádu tisíců řádků. Pro zjednodušení a zrychlení vývoje vzhledu se často používají preprocesory, frameworky, nebo různé knihovny. Vývojáři tak nemusejí opakovat se prvky vyvíjet od začátku.

V rozsáhlejších projektech s dlouhodobým trváním se však ukazuje, že použití frameworků není adekvátní. Často se ukazuje, že je velmi důležitá rychlost webových stránek (Lazaris, 2011), kterou mohou frameworky brzdit. Webové stránky navíc často rostou a správa CSS pak může být velmi náročná (Arsenault, 2017). Z tohoto důvodu vzniká řada metodik a přístupů ke tvorbě CSS stylů, které tento problém řeší. Mezi nejvýznamnější patří např. BEM, OOCSS, SMACSS, ATOMIC CSS. Všechny tyto přístupy mají některé společné vlastnosti, jako znovu použitelnost a modulární strukturování zdrojového kódu.

XML

Extensible Markup Language (XML)¹⁰ je značkovací jazyk vytvořený organizací W3C. Je to standard pro vytváření dokumentů pomocí značek a částečně i popisuje chování programů, které jej zpracovávají (W3C, 2008). V posledních letech je využíván v řadě odvětví lidské činnosti, např. právo, letectví, finance, pojišťovnictví, robotika, multimédia, pohostinství, cestovní ruch, umění, stavitelství, telekomunikace, softwarové inženýrství, zemědělství, fyzika, a mnoho dalších (Harold a Means, 2004). Hlavní výhody formátu XML jsou dle (W3Schools, 2017) následující:

- Odděluje data od HTML.
 - Pokud je třeba často měnit dynamická data.
- Zjednodušuje sdílení dat.
- Zjednodušuje přenos dat.
- Zjednodušuje změny platformy.
- Zlepšuje přístupnost dat.

XML vychází ze standardu SGML a je tak jeho aplikací. Přesněji XML dokumenty jsou v souladu s dokumenty SGML (W3C, 2008). V podstatě se jedná o textové dokumenty, editovatelné v jakémkoli textovém editoru. XML může používat dva druhy MIME type - „text/xml“ pro lidsky čitelnou podobu a v opačném případě „application/xml“. Základní prvky, které tvoří syntaxi jazyka XML jsou následující:

- **Značka (tag)** `<test>` (otevírací) `</test>` (uzavírací)
- **Atribut** `nazev="hodnota"` (nachází se uvnitř značek)
- **Text** mezi otevírací a uzavírací značkou (zde však mohou být i další elementy)
- **Element** – zahrnuje otevírací a uzavírací značku i s obsahem uvnitř
- **Entity** - reference na entity (symboly, znaky či jejich sekvence, např. ` `;))

Samotná syntaxe je však mnohem striktnější. Dokumenty, které jsou v souladu s touto syntaxí, se nazývají *Well-Formed*. Je pak logické, že pokud nejsou, nemohou být

¹⁰ <https://www.w3.org/TR/2008/REC-xml-20081126/>

nazývány XML dokumenty (Fawcett et al., 2012). Hlavní pravidla pro jejich tvorbu jsou následující (Eckstein a Casabianca, 2001; Harold a Means, 2004; Fawcett et al., 2012):

- Všechny elementy musí mít otevírací i uzavírací značku.
 - Pokud je značka prázdným elementem, musí být ukončen lomítkem (/).
- Všechny hodnoty atributů musí být uzavřeny v uvozovkách.
- Všechny atributy musí mít hodnotu.
- Elementy mohou obsahovat jiné elementy.
- Elementy se nesmí překrývat.
- Dokument nesmí obsahovat osamocené speciální znaky - & a <.
 - Musí být nahrazeny entitami.
- Je povolen pouze přesně jeden kořenový element.
- Element nesmí mít dva atributy stejného jména.
- Komentáře a procesní instrukce nesmí být uvnitř značek.
- Atributy dokumentů bez definovaného DTD jsou defaultně typu CDATA.

Důležitým pojmem, který se v souvislosti s XML často používá, je *validace*. XML je často používán jako metajazyk pro tvorbu dalších odvozených jazyků pro různé účely. Je tedy nutné tyto jazyky nějak definovat. Zároveň je potřeba aparát, který umožní validaci oproti těmto definicím. K tomu slouží několik technologií a aplikací XML. Mezi nejpoužívanější patří DTD (Document Type Definition) a XML Schema, méně pak RELAX NG a Schematron.

Další technologie související s XML je XSL (The Extensible Stylesheet Language). Ta je rozdělena na 2 části:

1. XSL Transformations (XSLT),
2. XSL Formatting Objects (XSL-FO).

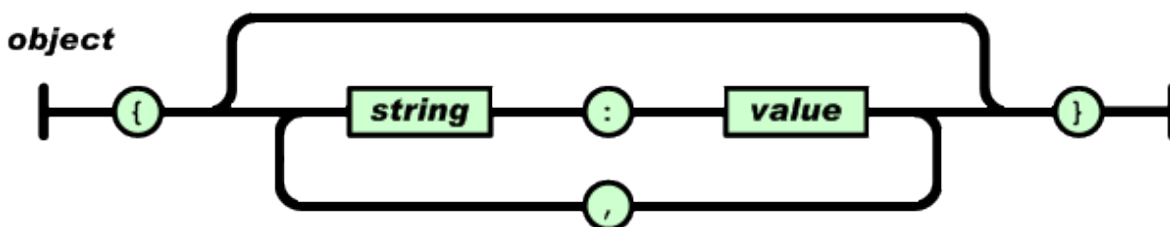
XSLT je aplikace XML, která se používá pro specifikaci pravidel pro transformování XML dokumentů do jiných XML. XSL-FO pak popisuje přesné rozložení textu na stránce, přičemž se jedná o stránku určenou k tisku (Harold a Means, 2004).

JSON

JSON (JavaScript Object Notation) je odlehčený textový formát pro výměnu dat definovaný ve standardu ECMA-404¹¹ (JSON.org, 2018). Jedná se o otevřený standard určený pro lidsky čitelnou, snadno generovatelnou a analyzovatelnou výměnu dat. Je odvozený z formátu JavaScriptových datových struktur. JSON je na JavaScriptu nezávislý a lze s ním pracovat v jakémkoliv programovacím jazyce. Původní specifikaci formátu JSON vytvořil Douglas Crockford (Crockford, 2006). Ačkoliv se jedná o textový typ, oficiální MIME type tohoto formátu je „*application/json*“.

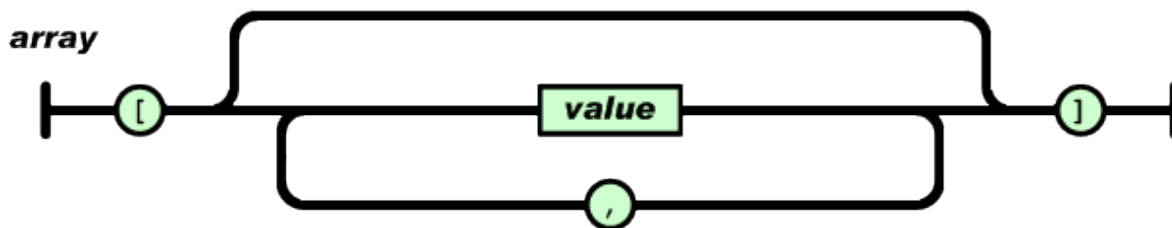
Formát JSON je vhodný pro serializaci dat a jejich přenos ve strukturované podobě prostřednictvím sítě. Nejčastěji se využívá při přenosu dat ze serveru ke klientovi a slouží tak jako alternativa k XML. Často se využívá při přenosu do mobilních zařízení, především díky jeho menší datové náročnosti oproti XML. Slouží také jako jeden z datových formátů pro AJAX.

JSON je založen na strukturách kolekce párů (název a hodnota) a tříděném seznamu (viz Obrázek č. 10 a Obrázek č. 11). Kolekci párů je možné realizovat jako objekt, záznam, strukturu, slovník nebo například asociativní pole. Tříděný seznam je možné reprezentovat v podobě pole, vektoru, seznamu nebo sekvence (JSON.org, 2018).



Obrázek č. 10 - Struktura syntaxe JSON - objekt; zdroj: (JSON.org, 2018)

¹¹ <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>



Obrázek č. 11 – Struktura syntaxe JSON - pole; zdroj: (JSON.org, 2018)

JavaScript

JavaScript je slabě typový, multiplatformní a objektově orientovaný skriptovací jazyk. Autorem původního návrhu je Brendan Eich ze společnosti Netscape. JavaScript podporuje objektovou orientaci, imperativní a funkcionální programovací styly. Jedná se původně o jazyk prototypovaný, čímž simuluje principy a vlastnosti třídově založených („class-based“) jazyků – například dědičnost (Stefanov a Sharma, 2013; ECMA, 2015). V současné době je označení „JavaScript“ obchodní známkou společnosti ORACLE AMERICA, INC. Standardizovaná verze JavaScriptu tak nese označení ECMAScript a je definována standardem ECMA-262¹² a současně ISO/IEC FDIS 22275¹³. Dle Stefanova a Sharma (2013) je pak *JavaScript* to, co je implementováno všemi prohlížeči a prostředími.

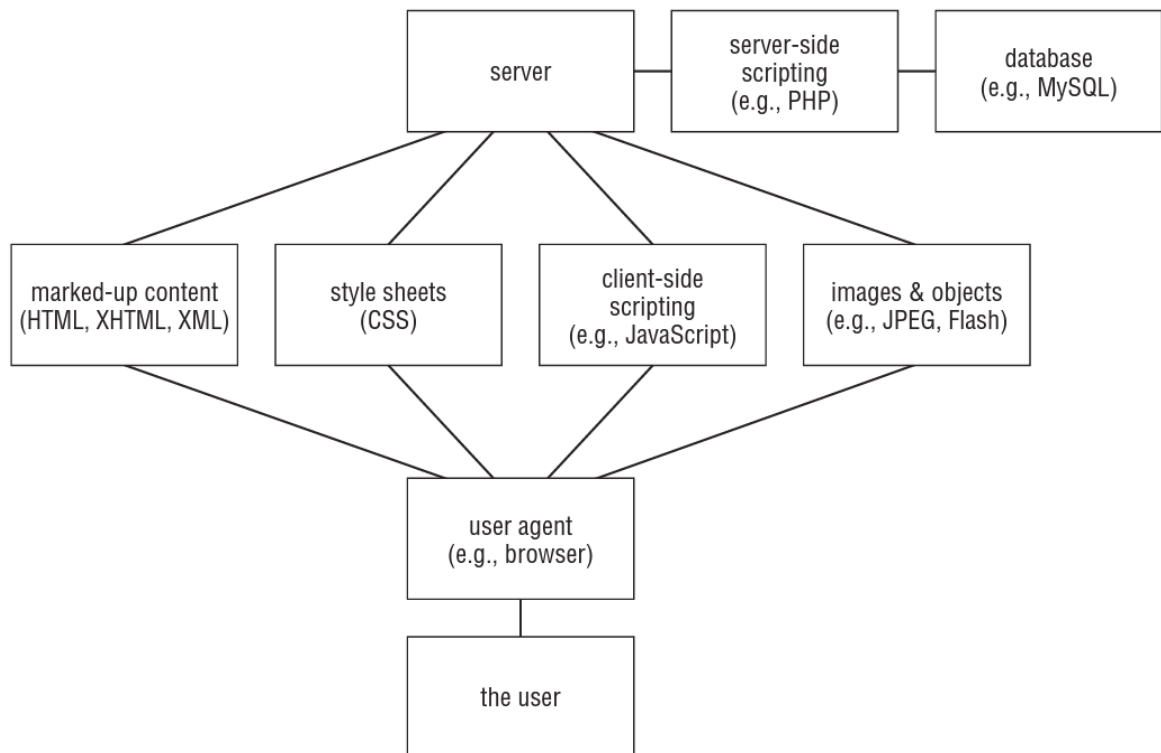
Zatím nejběžnější použití JavaScriptu je v podobě interpretovaného programovacího jazyka na webových stránkách. Je možné jej vložit do stránky několika způsoby:

1. jako interní skript,
2. v podobě externího skriptu,
3. formou řádkového zápisu události.

Existují ještě další možnosti, jak může být webová stránka doplněna o JavaScript – například jeho dynamickým načtením ze serveru pomocí AJAXu.

¹² <https://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>

¹³ <https://www.iso.org/standard/73002.html>



Obrázek č. 12 - JavaScript v modelu client-server; zdroj: (Goodman, 2010)

Skripty na webových stránkách pracují s objektovým modelem dokumentu DOM (Document Object Model). DOM je platformě nezávislé rozhraní, které umožňuje dynamicky přistupovat a upravovat obsah, strukturu a styl dokumentů (Goodman, 2010; W3C, 2005). JavaScript tak zajišťuje funkcionality webových aplikací a interaktivitu webových stránek. Syntaxe jazyka je odvozena od rodiny jazyků C (C, C++, Java). Samotné slovo Java bylo společností Netscape použito v názvu pouze z marketingových důvodů.

JavaScript, nebo spíše ECMAScript je velmi rychle rozvíjející se jazyk. Webovým prohlížečům tak často trvá nějakou dobu, než převezmou nejnovější prvky tohoto jazyka. ECMAScript se tak často překládá do verze, která je kompatibilní s většinou webových prohlížečů. Tento druh překladač se nazývá transpilování. Přesnější definici nabízí (Sengstacke, 2016): „*Transpilers, or source-to-source compilers, are tools that read source code written in one programming language, and produce the equivalent code in another language.*“

V poslední době se často využívá JavaScript i na straně serveru, především v souvislosti s aplikačním serverem *Node.js* (Malý, 2010). Více je o této problematice pojednáno v kapitole 4.4.2.

4.4.2 Technologie server-side

PHP

Původní význam zkratky PHP byl Personal Home Page. Dnes je zkratka dle (PHP GROUP, 2015) uváděna jako rekurzivní akronym PHP: Hypertext Preprocessor. PHP je preprocesor, který vykonává stejnojmenný skriptovací jazyk.

PHP je tedy hypertextový preprocesor, který na serveru interpretuje stránky HTML s vlastními příkazy před jejich odesláním ke klientovi (obvykle je jím webový prohlížeč). To znamená, že PHP umožňuje vkládat vlastní skripty (krátké úseky kódu, ale i celé programy) přímo do hypertextových stránek (Bráza, 2005). Tyto skripty se vkládají do HTML pomocí speciálních značek (`<?php a ?>`). Aby se odlišily soubory, které mají být zpracovány PHP preprocesorem, užívá se pro ně textových souborů s příponou `*.php`.

PHP jako skriptovací (programovací) jazyk je uzpůsoben pro webový vývoj a práci s HTML (PHP GROUP, 2015). Řadí se mezi objektově orientované a interpretované jazyky. Syntaxe je velmi podobná jazyku C.

Mezi další klíčové vlastnosti jazyka PHP patří vágnost typování proměnných. V jazyce není třeba uvádět typy proměnných, ani proměnné inicializovat. Inicializace se provede dynamicky při prvním použití. Zároveň proměnné není třeba likvidovat, to se provede automaticky. Důležitá je snaha o kompaktnost kódu. Zápisy mohou být často jednořádkové a jazyk taktéž dovoluje několikanásobné vrstvení funkcí.

PHP má modulární architekturu a umožňuje tak připojovat knihovny s rozšiřujícími funkcemi. Díky tomu dokáže spolupracovat s řadou databázových systémů.

Vedle PHP se na straně serveru používá ještě několik programovacích jazyků a systémů. Jedná se především o:

- framework .NET (ASP.NET) a MVC pro webové aplikace,
- jazyk Ruby a framework Ruby on Rails,
- Java Server Pages.

Node.js

Node.js je JavaScriptové běhové prostředí, založené na Chrome V8 engine (Node.js Foundation, 2015). Původně jej navrhl Ryan Dahl v roce 2009. Od té doby můžeme sledovat rychlý vzestup této platformy a její využívání při vývoji webových a mobilních aplikací.

Používá se obecně pro tvorbu webových aplikací s vysokými nároky na výkon, real-time zpracování a škálovatelnost. Konkrétně jde o vývoj jednostránkových webových aplikací, hybridní mobilní aplikace, různé API, či aplikace pro zpracování dat pro internet věcí.

Platforma Node.js funguje na odlišném principu než PHP + Apache, nebo .NET + IIS. Architektura Node.js je postavená na událostech (event-driven). Základní principy jsou dle Pasqualiho (2013) následující:

- Node program/proces běží na jednom vlákně – event loop.
- Předpoklad intenzivních I/O operací.
- Běh programu je řízen pomocí asynchronních callbacků.
- Operace využívající hodně procesorového času jsou odděleny do samostatného procesu a vysílají události, když mají výsledky.
- Komplexní programy by měly být sestaveny z menších jednodušších programů.

Pro vývoj aplikací se využívá programovací jazyk JavaScript (resp. ECMAScript). To má výhodu v tom, že vývoj na straně serveru i klienta využívá stejného jazyka se stejnou syntaxí.

Node.js dokáže pracovat s řadou databázových serverů, ale nejčastěji se používá ve spojení s NoSQL databázemi, např. MongoDB.

Databáze

Na straně serveru je nutné vhodným způsobem ukládat data. Agendové zpracování dat nahradily databáze a ty se používají i v prostředí WWW. Nejčastěji se využívají relační databázové systémy, které v různé podobě podporují jazyk SQL. V poslední době se ale objevují i tzv. NoSQL databáze.

Nejrozšířenějším relačním databázovým systémem využívaným pro chod webových stránek je MySQL. Mezi další využívané systémy patří Microsoft SQL Server, PostgreSQL a okrajově také souborová SQLite.

Relační databázový systém umožňuje uživatelům vytvářet jednotlivé samostatné databáze, přičemž každá databáze může obsahovat jednu či více tabulek. Tabulky jsou tvořeny řádky a sloupci. Řádky odpovídají jednotlivým záznamům, datovým větám, sloupce atributům, jednotlivým prvkům těchto datových vět. To znamená, že všechny datové věty v jedné tabulce mají stejnou strukturu atributů, danou strukturou tabulky. Každý atribut má předem určen datový typ, jakého mohou hodnoty nabývat (Gilmore, 2005).

NoSQL databázové systémy nevyužívají široce rozšířený relační datový model (a RDBMS) a tudíž ani jazyk SQL (Vaish, 2013). Pojem však neoznačuje typ databáze, ale v zásadě jen to, že se v nich nevyužívá RDBMS a SQL. Hlavním důvodem vzniku NoSQL databází je škálovatelnost. Ta je v případě relačních databází omezená. Mezi další důvody pak patří prezentace dat bez použití schémat, rychlejší čas vývoje, výkon a další (Vaish, 2013).

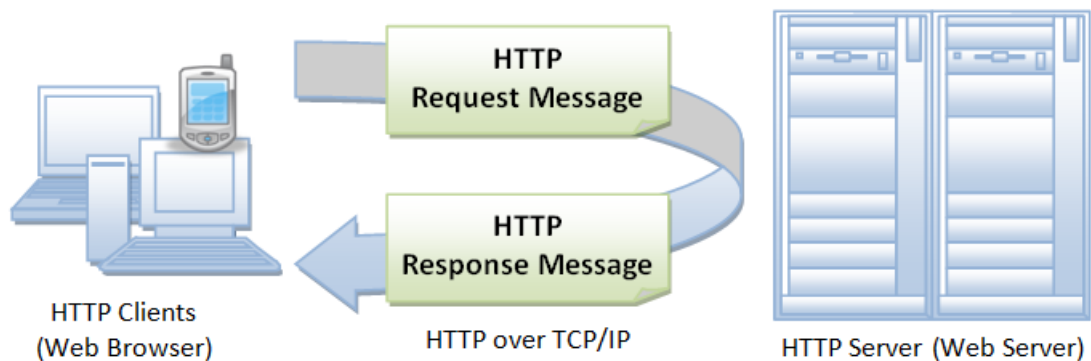
Nejrozšířenějším NoSQL databázovým systémem je MongoDB. Ten se používá především ve spolupráci s Node.js. Řadí se mezi dokumentové databáze a data ukládá do dokumentů ve formátu, který je podobný objektům v JSON (MongoDB Inc., 2015).

Při vývoji moderních, nejen webových, aplikací se často využívají různé databázové vrstvy. Ty abstrahují práci s databázovými systémy a vývojářům dávají možnost výběru, který systém využijí. Je tak možné již u běžících aplikací změnit databázový systém s minimálním množstvím úprav zdrojového kódu.

4.4.3 Protokol HTTP

O přenos dat mezi klientem a serverem se stará Hyper Text Transfer Protocol (HTTP). První návrh představil již Tim Berners-Lee spolu s prvním návrhem webu. Dnes je pod správou konsorcia W3C.

HTTP je považován za bezstavový protokol. Jednotlivé příkazy jsou prováděny nezávisle na sobě. Není tak nic známo o příkazech, které byly provedeny před nimi. Přenos dat mezi klientem a serverem probíhá v textovém tvaru a je založen na principu požadavek - odpověď. Schematicky znázorňuje komunikaci Obrázek č. 13. Důležitou součástí jsou stavové kódy (tzv. *response status codes*), které určují povahu odpovědi serveru. Podle typu jsou rozdělené do pěti kategorií (1xx, 2xx, 3xx, 4xx, 5xx).



Obrázek č. 13 - Základní komunikace pomocí protokolu HTTP; zdroj: (Hock-Chuan, 2009)

Aktuálně se používá verze HTTP/1.1, která byla vydaná již v roce 1999. V současné době je již hotová verze HTTP/2. Vydaná je pod Internet Engineering Task Force (IETF) jako RFC 7540¹⁴. V současné době je využíváno přibližně 23,5% webů (W3Techs, 2018). Vylepšení jsou především v oblasti výkonu a reflektují změny v moderních technologiích a jejich požadavky na přenos dat.

¹⁴ Request for comments; <https://tools.ietf.org/html/rfc7540>

4.4.4 AJAX

AJAX (Asynchronous JavaScript and XML) představuje, spíše než technologii, přístup ve vývoji webových aplikací (Ford, 2009). Kombinuje stávající technologie a standardy. Obecně lze říci, že AJAX dovoluje, aby byly stránky aktualizovány asynchronně změnou malých objemů dat na pozadí. To znamená, že je možné aktualizovat části webové stránky bez nutnosti obnovení celé stránky (W3Schools, 2015).

Za výhody použití technologie AJAX je považována úspora načítaného objemu dat na stranu klienta a přiblížení se aplikace svým rozhraním aplikacím desktopovým – jedná se o tzv. RIA („Rich Internet Applications“). Nevýhodou pak může být větší počet HTTP požadavků, který při nižších rychlostech nebo dobách odezvy internetového připojení může značně komplikovat činnost internetových aplikací.

Pravděpodobně první zmínka o AJAXu pochází od Garretta (2005). Ten definuje AJAX následovně:

„Ajax isn't a technology. It's really several technologies, each flourishing in its own right, coming together in powerful new ways. Ajax incorporates:

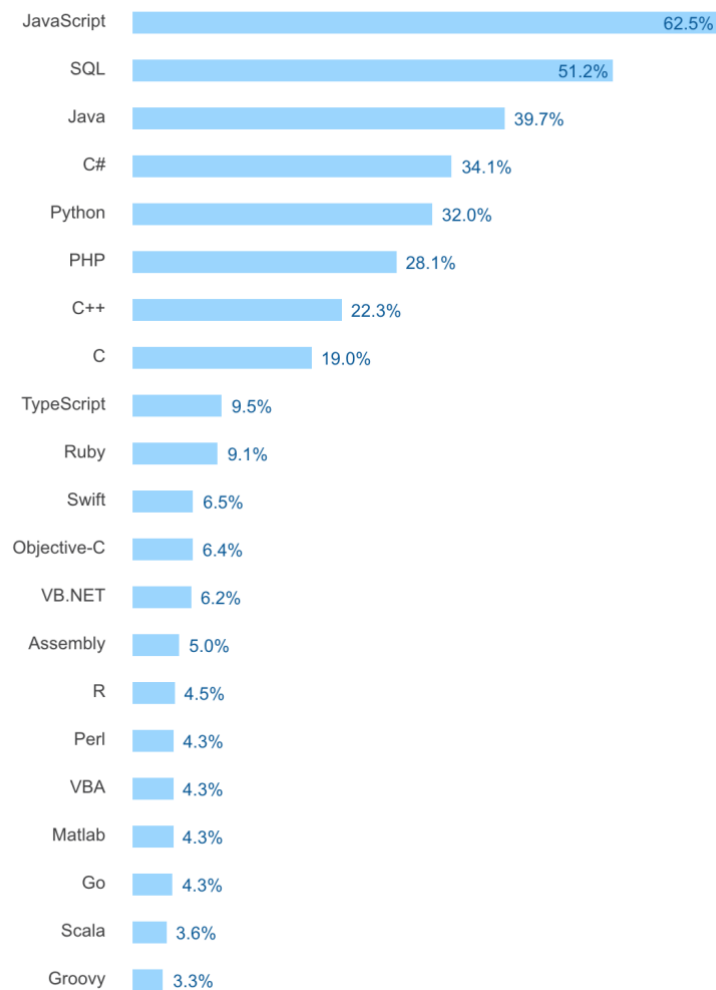
- *standards-based presentation using XHTML and CSS;*
- *dynamic display and interaction using the Document Object Model;*
- *data interchange and manipulation using XML and XSLT;*
- *asynchronous data retrieval using XMLHttpRequest;*
- *and JavaScript binding everything together.“*

Podstata AJAXu se od té doby nezměnila. Použité technologie se postupně vyvíjejí. V dnešní době se tak místo XHTML používá HTML5. Objekt XMLHttpRequest nyní spravuje a vytváří pro něj specifikaci organizace W3C. Přenášená data nejsou dnes pouze ve formátu XML, ale mohou být v prostém textu, nebo formátu JSON (Ford, 2009).

Většina vývojářů dnes nepotřebuje znát přesné fungování AJAXu (např. práci s objektem XMLHttpRequest). Tuto funkcionalitu dnes obstarávají frameworky a knihovny na straně serveru, nebo klienta.

4.4.5 Moderní trendy a přístupy k vývoji aplikací

Se stále rostoucí dynamikou webových aplikací a stále intenzivnějším používání AJAXu roste objem kódu v jazyce JavaScript vykonávaného ve webovém prohlížeči. Díky tomu se vyvinuly pokročilé frontendové frameworky jako Angular, React nebo Ember. S jejich pomocí se nejčastěji vyvíjí tzv. single-page aplikace. JavaScript jako programovací jazyk je stále populárnější a čím dál více aplikací jej využívá. To dokládají např. statistiky známého vývojářského fóra stackoverflow.com – viz Obrázek č. 14.



Obrázek č. 14 - Statistiky užívání programovacího jazyka uživateli (programátory) webového fóra stackoverflow.com; zdroj: (Stack Overflow, 2017)

Single-page aplikace

Koncept single-page aplikací (SPA) opouští tradiční pojetí webových stránek, kde se při přechodu na jinou stránku na základě http požadavku přepisuje celá stránka. Malý (2011) je definuje jako aplikace, kde je veškerá funkcionalita umístěna v jedné stránce a se serverem se pracuje jen prostřednictvím datového rozhraní.

Značná část logiky je v těchto aplikacích přenesena ze serveru na klienta. Při přechodu mezi jednotlivými stránkami tak není načítána celá stránka, ale pouze ty části, které se mění. Aplikace tedy v prohlížeči řeší i samotný routing a změnu adresy URL v prohlížeči.

Izomorfní aplikace

V souvislosti s rozvojem Node.js a tedy možností psát serverovou část stejným jazykem jako klientskou se uchytil pojem *izomorfní aplikace*. Tento přístup byl prvně popularizován Spikem Brehmem ze společnosti Airbnb ve spojitosti s rozvojem výše zmíněných single-page aplikací. Výsledné HTML bylo původně renderované až na straně klienta. To sebou neslo problém s nemožností indexace webových stránek vyhledávači. Izomorfní přístup pak využívá možnosti běhu JavaScriptu na serveru v rámci Node.js. Node.js tedy umožňuje spouštět aplikace již na serveru a odeslat do klientovi již vyrenderované HTML (Brehm, 2013). To jsou pak schopny přečíst i indexovací nástroje vyhledávačů.

Funkcionální programování

V prostředí WWW je vidět vzrůstající popularita využívání funkcionálního přístupu k programování. Např. Vogels (2017) jej řadí mezi hlavní trendy ve webovém vývoji v roce 2018. Definici moderního pojetí funkcionálního programování nabízí např. Jenkins (2015): *„Functional programming is about writing pure functions, about removing hidden inputs and outputs as far as we can, so that as much of our code as possible just describes a relationship between inputs and outputs.“*. Seeman (2014) pak vysvětluje důvod odklonu od klasického objektového programování na základě SOLID

principů: „If you take the SOLID principles to their extremes, you arrive at something that makes Functional Programming look quite attractive.“

4.5 Sémantika v prostředí WWW

Problematika sémantiky je velice široké téma a samostatná vědní disciplína. Následující text se tak zaměřuje na oblast sémantiky relevantní k zaměření práce, tedy v prostředí WWW.

Konsorcium W3C pro tuto oblast používá pojem *Semantic web* a je součástí vize *Web of linked data* (W3C, 2015c). Sémantický web je nová forma webu, které rozumí i počítače (Berners-Lee et al., 2001). Sémantika je nejdůležitější pro vyhledávače, chytré asistenty, prohlížeče, sociální sítě, obecně nástroje indexující obsah a v neposlední řadě také rozvíjející se obor umělé inteligence (AI, artificial intelligence). Sémantika tak pomáhá počítačům a jejich programům porozumět a zpracovávat informace.

V současné době, je velmi populární pojem umělá inteligence. Tento pojem je veřejností značně nadhodnocen a současné systémy se umělé inteligenci ještě příliš neblíží. Programy spadající pod tuto oblast zatím nejsou schopny příliš rozumět lidskému jazyku a tedy obsahu samotnému. Sémantika je tak pro ně klíčová (Hendler a Berners-Lee, 2010). Dnešní chytré asistenty, jako jsou Google Home, Amazon Echo, Siri apod., které jsou za nás schopné vyhledat informace na webu, nakoupit, zarezervovat cokoli, potřebují ke své funkci právě sémantiku (Sweney, 2016).

4.5.1 Sémantika a HTML

K jednotlivým elementům, atributům a hodnotám atributů definuje specifikace HTML5 význam obsahu – tedy toho, co může být uvnitř (W3C, 2017b). Samotná specifikace je však v tomto ohledu velmi obecná. Umožňuje však využití dalších nástrojů, k čemuž dává vývojářům přímé možnosti.

Jedním z těchto doplňujících nástrojů je framework WAI-ARIA. Ten přiřazuje hodnoty atributům *role* a *aria-**. Rozšiřují tak význam jednotlivých elementů

a pomáhají v oblasti přístupnosti především s dynamickým obsahem a pokročilými prvky uživatelského rozhraní využívající JavaScript a AJAX (Web Accessibility Initiative, 2016).

Pro informační obsah je, jak již bylo zmíněno v předchozím textu, nejdůležitější to, co je možné umístit uvnitř elementu *article*. Pro sémantiku v prostředí WWW je pak nejdůležitější metadatový popis.

4.5.2 Metadatový popis

Pro rozšíření základní sémantiky v HTML je možné připojovat metadata a metadatový popis. Metadata se nejčastěji zjednodušeně definují jako „data o datech“, nebo „informace o informacích“. Existuje ale i celá řada dalších, komplexnějších definic. Např. definici blízko zaměření práce je možné nalézt v Berners-Lee (1997): „*Metadata is machine understandable information about web resources or other things.*“

Pro popis metadaty existuje řada formátů. Specifikace HTML5 definuje přímo syntaxi pro tzv. Microdata. Jedním z nejstarších formátů je pak RDF a RDFa. Naopak nejnovějším formátem je JSON-LD. Tyto tři formáty pak využívá i největší vyhledávač Google ve své koncepci *Structured data*¹⁵ pro lepší porozumění obsahu při indexování. Další použitelnou formou jsou ještě Mikroformáty. Ty využívají z HTML pouze atribut *class*. Další metadatový popis je možné umístit do hlavičky HTML dokumentu dovnitř, k tomu určenému, elementu <meta>. Patrně nejrozšířenějším formátem, vedle základu definovaného v HTML5, je v této oblasti OpenGraph protokol, který využívají sociální sítě.

Microdata

W3C (2017c) definuje Microdata následovně: „*This mechanism allows machine-readable data to be embedded in HTML documents with an unambiguous parsing model. It is compatible with JSON, and can be written in a style which is convertible to RDF.*“

¹⁵ <https://developers.google.com/search/docs/guides/intro-structured-data>

V HTML5 jsou microdata tvořena hodnotami atributů *itemscope*, *itemprop*, *itemref*, *itemtype* a *itemid*. V základu jsou definována jako skupiny dvojic název-hodnota. Tyto skupiny jsou nazývány *items* a označeny atributem *itemscope* (bez hodnoty). Dvojice název-hodnota je označena atributem *itemprop* a obsahem elementu, ve kterém se nachází (případně hodnotou atributu *value*). Například:

```
<div itemscope>
|   <p>My name is <span itemprop="name">Daniel</span>.</p>
| </div>
```

Zdrojový kód č. 1 - Ukázka zápisu metadatového formátu Microdata do HTML; zdroj: vlastní zpracování

Specifikace HTML Microdata definuje především syntaxi a zpracování dat. Pro správnou interpretaci je nutné pro každou skupinu (*itemscope*) definovat typ. Typ je určen slovníkem. K tomu se používá atribut *itemtype*. Slovníky pak definují objekty a jejich strukturu.

RDF a RDFa

RDF (Resource Description Framework) je nejstarší soubor specifikací, které vypracovalo konsorcium W3C. RDF byl původně navržen jako model metadat a slouží jako obecná metoda pro modelování informací v různých syntaxích (W3C, 2009).

V rámci RDFa (Resource Description Framework in Attributes) pak W3C definuje soubor několika specifikací, které určují způsob zápisu informací z RDF do HTML a XHTML dokumentů pomocí atributů (W3C, 2015b).

Obdobně jako Microdata, RDFa specifikuje především způsob zápisu a serializace. RDFa se tak na webu často používá se slovníkem Schema.org.

JSON-LD

JSON-LD je stejně jako předchozí nástroj pro strukturovaný popis dat. Používá formát JSON a byl vyvinut pro systémy, které již JSON používají. Umožňuje rozšířit

stávajících aplikací a snadné ukládání a implementaci do moderních vývojových prostředí (W3C, 2014c).

JSON-LD je možné integrovat s minimálním zásahem do existujícího HTML. Zapisuje se v definovaném formátu JSON do elementu script (`<script type="application/ld+json">`).

Všechny zmíněné formáty (JSON-LD, RDFa i Microdata) jsou při správném zápisu mezi sebou převoditelné.

Metadatové slovníky

Výše zmíněné metadatové formáty se vždy používají ve spojení se slovníkem (*metadata vocabulary*), který definuje a popisuje objekty a dává jim význam. Přesnou definici nabízí Bertino et al. (2008): „*A formal definition of a set of descriptors to be used for denoting the characteristics of resources (e.g., an ontology is a metadata vocabulary). Usually, metadata vocabularies are domain-specific.*“

V prostředí WWW se patrně nejvíce používají slovníky iniciativy Schema.org, která byla založena společnostmi Google, Microsoft, Yahoo a Yandex (Schema.org, 2017). Společnost Google jej pak podporuje ve svém vyhledávači v rámci koncepce *Structured data*. Dalším populárním slovníkem je např. Dublin Core¹⁶. Přesné statistiky však neexistují.

Mikroformáty

Podobně jako předchozí, jsou Mikroformáty způsob zápisu strukturovaných informací do HTML. V rámci jazyka se k pojmenování využívá atribut class a jeho hodnoty. Data z mikroformátů lze také převést do kanonického JSON (microformats2, 2017). Na rozdíl od formátu Microdata, Mikroformáty specifikují i metadatový slovník, a tedy i objekty, které je možné používat. Nelze tak použít slovník Schema.org.

¹⁶ <http://www.dublincore.org/>

Open Graph

Nejdůležitějším metadatovým formátem pro sociální sítě a zejména Facebook je *Open Graph* protokol. Ten umožňuje jakékoli webové stránce stát se objektem tzv. „social graph“ a mít tak stejnou funkcionalitu jako kterýkoli jiný objekt na Facebooku (Facebook, 2017). Používá se na úrovni stránek a pro zaměření práce je důležitý především pro správný metadatový popis stránky při sdílení na sociálních sítích. Ty jsou velmi důležitým zdrojem návštěvnosti zpravodajských webů.

Metadata formátu Open Graph protokol se umísťují do elementu `<head>` v hlavičce dokumentu a využívají elementy `<meta>`, pro které definují syntaxi hodnot atributů *property* a *content*.

4.6 Shrnutí

Nejprve byl na základě pojmů data, informace, znalosti a prostředí WWW vymezen pojem informační obsah pro další využití v této práci.

Základní technologií pro zobrazení webových stránek a aplikací je jazyk HTML. Dnes se jednotlivé technologie nepoužívají samostatně. Jazyk HTML je doplněn o interaktivitu pomocí JavaScriptu, často se komunikuje se serverovou částí pomocí AJAXu. Ten pak využívá protokol HTTP a jazyk XML, nebo JSON.

Vývojáři dnes málokdy vyvíjejí webové stránky a aplikace pouze s využitím základních technologií. Pro urychlení vývoje se tak používají různé frameworky. Pro samotný design stránek jsou to především CSS frameworky jako Bootstrap, Foundation apod. Ty využívají jak samotné kaskádové styly, tak kód v jazyce JavaScript. Pro složitější, především tzv. single-page aplikace, existují zvláštní frameworky. Mezi nejznámější patří Angular.js, Ember, nebo React.js.

Nejvyužívanější technologií na straně serveru je v současné době stále PHP. Vedle něj je velmi využívanou technologií platforma .NET od Microsoftu. Jde o profesionální

technologii, která je v řadě ohledů vyspělejší než PHP. Širšímu rozšíření však brání uzavřenost platformy a nutnost provozu operačního systému Windows na serverech.

V poslední době se díky rozmachu mobilních zařízení objevují nové serverové systémy. Ty reagují na odlišné požadavky a zvyšující se počet zařízení, která je nutné obsluhovat. Nejviditelnější je v této oblasti Node.js.

I na straně serveru se používá řada frameworků. Pro PHP jsou nejznámější Symfony, Laravel a v České republice Nette. Pro jazyk Ruby je to Ruby on Rails a pro .NET MVC 5, pro Node.js pak např. Express, Derby, SocketStream, Meteor nebo LoopBack.

V současné době je v prostředí WWW vidět odklon od klasického pojetí webových aplikací, kde je logika aplikace rozdělena mezi klienta a server. S rozvojem single-page aplikací se většina logiky přesouvá do webového prohlížeče a se serverem se komunikuje jen s pomocí datového rozhraní. Čím dál více vývojářů využívá pro aplikace jazyk JavaScript a odklání se od tradičního PHP. Je otázkou, jak moc tento trend bude v budoucnu pokračovat.

Velmi důležitou problematikou v prostředí WWW je sémantika. Se stále rostoucí oblastí automatických systémů a umělé inteligence její význam neustále roste. Pro sémantiku a Sémantický web, jak jej nazývá konsorcium W3C, je nezbytný metadatový popis. Nejpoužívanějšími formáty v prostředí webu jsou Microdata, RDFa a JSON-LD.

Microdata a RDFa zasahují ze své podstaty do HTML. JSON-LD je naproti tomu zaměřený na existující systémy a využívá velmi populární formát JSON. Iniciativa Schema.org v současné době podporuje jak Microdata, tak i RDFa a JSON-LD. Alternativní Mikroformáty na rozdíl od těchto využívají jen omezený slovník, který je definován v rámci specifikace. Navíc nejsou přímo podporovány konsorciem W3C.

5 Metody a nástroje pro sběr a zpracování dat v prostředí WWW

5.1 Správa informačního obsahu v prostředí WWW

Díky rozvoji internetových technologií, zejména redakčních systémů, mohou vkládat a aktualizovat informační obsah webových stránek i lidé bez odborných znalostí konkrétních technologií - HTML, CSS, JavaScript (Brown, 2014). V praxi se většinou jedná o vkládání příspěvků (články, aktuality, rozhovory, reportáže a další), kde hlavní část obsahu je často řešena pomocí tzv. WYSIWYG (What You See Is What You Get) editorů. Na to jsou navázány další nástroje redakčních systémů a další obsahové prvky, jako např. fotografie, odkazy na související příspěvky, souborové přílohy, externí odkazy apod. WYSIWYG editory umožňují pracovat s obsahem bez nutnosti znalosti zmíněných technologií, podobně jako v běžných textových editorech (např. MS Word). To však s sebou nese řadu omezení.

Neznalost problematiky značkovacího jazyka HTML a jeho napojení na kaskádové styly (CSS) způsobuje značně ztíženou práci s pokročilými nástroji a vlastnostmi (např. obtékání pomocí CSS vlastnosti *float*, rozložení prvků pomocí *flexbox* apod.). WYSIWYG nástroje nejsou dokonalé, výsledný obsah často neodpovídá zamýšlenému (Komenda, 2017) a vznikají zde problémy s konzistencí HTML kódu. Tuto problematiku je pak nutné řešit dalšími programovými aparáty (Spiesser a Kitchen, 2004). Dále je nutné jejich napojení na další prvky webu (např. redakčního systému). Velkým problémem je pak responzivita výsledného obsahu. Tu současné WYSIWYG editory neřeší vůbec.

Digitální podoba médií pak poskytuje řadu možností, které se dají využít (Das et al., 2009). Obsah (text) je možné doplňovat o řadu dalších prvků, jako jsou obrázky s popisy, fotogalerie, videa, různé podoby odkazů, interaktivní mapy, interaktivní prvky obecně a další.

Dále je nutné řešit problematiku ukládání obsahu, především z hlediska formátu uložení. Současné redakční systémy používají WYSIWYG editory pro hlavní část

informačního obsahu, který lze doplnit dalšími bloky. Většina redakčních systémů a jejich rozšíření ukládají bloky obsahu v samostatných tabulkách, nebo v attributech tabulky. Celkově pak všechna existující řešení ukládají obsah v různých formátech (strukturách). To způsobuje problémy při majoritních aktualizacích redakčních systémů, nebo při přechodu na jiný.

Dále je nutné podporovat moderní technologie a zařízení jako jsou chytré mobilní telefony, tablety, televize a podobně. Často je potřeba obsah předávat do různých aplikací v jiném formátu, než je využíván při prezentaci na webu.

5.2 Web Content Management System











Pojem Web Content Management System je obecný název pro webové aplikace, které se využívají pro chod a správu webových stránek. V češtině se obecně nazývají redakční, nebo také publikační systémy (v angličtině často zkráceně Content Management Systems). Existuje velké množství redakčních systémů, které se dále dělí do řady kategorií dle účelu využití. Liší se svojí složitostí, určením, mírou univerzálnosti užití apod. Portál opensourcecms.com jich v současné době (2015/2016) eviduje přes 320.

Původně byl pojem Content Management System využíván v oblasti technických dokumentací a mezi vědeckými komunitami. Základním problémem bylo znovuvyužití částí textů (Boiko, 2005). Redakční systémy, jak je známe dnes, se objevily až s příchodem prostředí World Wide Web. Vznikly především pro zjednodušení správy webových stránek a portálů. K tomu patří i správa samotného obsahu. Postupem času, jak se redakční systémy vyvíjejí, je tato správa jednodušší. Webové stránky tak mohou spravovat i lidé bez odborných znalostí webových technologií.

Definici redakčního systému nabízí Seadle (2006): „*A content management system (CMS) offers a way to manage large amounts of web-based information that escapes the burden of coding all of the information into each page in HTML by hand.*“

Mezi nejvíce využívané redakční systémy se dlouhodobě řadí Drupal, Joomla s WordPress. Statistiky se v přesných hodnotách využití rozcházejí, ale v pořadí

se shodují. Např. dle websitesetup.com situaci ke konci roku 2017 znázorňuje Obrázek č. 15.

#	WEBSITES USING	MARKET SHARE %	ACTIVE SITES	# OF WEBSITES IN MILLION
1	 WordPress	59.9 %	26,701,222	239,139
2	 Joomla	6.6 %	2,009,717	13,480
3	 Drupal	4.6 %	964,820	23,330
4	 Magento	2.4 %	372,915	12,095
5	 Blogger	1.9 %	758,571	15,779
6	 Shopify	1.8 %	605,506	11,587
7	 Bitrix	1.5 %	200,210	3,925
8	 TYPO3	1.5 %	582,629	3,568
9	 Squarespace	1.5 %	1,390,307	9,799
10	 PrestaShop	1.3 %	262,342	2,099

Obrázek č. 15 - Obsazení trhu redakčních systémů; zdroj: (Mening, 2017)

Nejpoužívanější redakční systémy jsou technologicky stále založeny na serverové technologii PHP ve spojení s relačním databázovým systémem. Nejčastěji se využívá databáze MySQL, ale je možné využít i jiné systémy (např. SQLite, nebo PostgreSQL). Klientská část zpravidla již využívá nejnovější HTML5, ale díky modularitě redakčních systémů záleží především na použité šabloně. Dynamika a AJAXovová komunikace je zde řešena většinou pomocí JavaScriptového frameworku jQuery.

Pomalu se však začínají objevovat i redakční systémy založené čistě na jazyce JavaScript, využívající moderní izomorfní přístup a často také NoSQL databáze. V současné době (začátek 2018) mezi nejznámější patří např. KeystoneJS¹⁷, Ghost¹⁸, ApostropheCMS¹⁹, nebo Total.js²⁰. Jejich funkcionality však nedosahují možností velkých redakčních systémů zmíněných výše.

¹⁷ <http://keystonejs.com/>

¹⁸ <https://ghost.org/>

¹⁹ <http://apostrophecms.org/>

²⁰ <https://www.totaljs.com/cms/>

5.3 Redakční systémy a správa obsahu

Redakční systémy v současné době ukládají obsah různým způsobem. Všechny však využívají WYSIWYG editor pro vytvoření hlavní části obsahu, doplněný o další přílohy (obrázky, dokumenty, video). Některé editory jsou rozšířeny o možnosti vkládání příloh přímo. Výsledný obsah je však příliš jednoduchý a jednotlivé části strojově obtížně oddělitelné a zpracovatelné. V poslední době redakční systémy řeší problém pomocí kompozice z nezávislých bloků obsahu. V následujících kapitolách je rozebráno, jak k tvorbě obsahu přistupují tři nejvýznamnější open source redakční systémy dle podílu na trhu (viz. Obrázek č. 15).

5.3.1 Drupal

Redakční systém Drupal je postaven na PHP frameworku Symfony. Celkově je velmi otevřený a přátelský pro vývojáře rozšíření. Je nejvíce technicky pokročilý, poskytuje mnoho možností pro úpravy, zároveň ale jeho administrace vyžaduje pokročilejší znalosti (Mening, 2015).

Jádrem redakčního systému Drupal pro práci s obsahem jsou tzv. *nody*. Téměř každý obsah je node. Každý node je pak složen z bloků (polí – *fields*). Ty lze pro každý typ obsahu konfigurovat (přidávat, odebírat, měnit pořadí). Pole je část obsahu, která může mít různou formu – např. text s WYSIWYG editorem, soubor, obrázek, odkaz atd. Je tak předem vytvořená struktura (typy polí, pořadí), která je pevná pro veškerý vytvářený obsah tohoto typu. Její změna se projeví u již vytvořeného obsahu a může dojít i ke ztrátě dat. Výjimku tvoří pole, která mohou být u obsahu ve více instancích. Editaci takové struktury ukazuje Obrázek č. 16.

[Show row weights](#)

FIELD	LABEL	FORMAT	
+ Image	- Hidden -	Image	Image style: Large (480x480) ⚙
+ Body	- Hidden -	Default	
+ testcontent	Above	Default	
+ Tags	Above	Label	Link to the referenced entity ⚙
+ Comments	Above	Comment list	⚙
+ Links		Visible	
Disabled			
<i>No field is hidden.</i>			

Obrázek č. 16 - Editace šablony polí obsahu; zdroj: vlastní zpracování

Výše zmíněné funkcionality jsou v redakčním systému Drupal obsaženy již v základní instalaci a tvoří nedílnou součást jádra celého systému. Dostupná rozšíření (moduly) tuto funkcionalitu dále rozšiřují. Pro pokročilou správu obsahu je k dispozici především modul Paragraphs²¹, který výrazným způsobem rozšiřuje možnosti a práci s poli obsahu. Přidává možnost dynamicky přidávat jednotlivá pole a vytvářet šablony pro standardní typy obsahu. Pořadí takových polí pak lze pro každou instanci obsahu měnit zvlášť. Možnosti modulu Paragraph pak ukazuje na Obrázek č. 17. Administrátoři však mají omezenou možnost změnit strukturu generovaného HTML pro prezentaci obsahu. Zároveň je velmi omezená práce se strukturováním textu a úrovní nadpisů, což je jedna ze základních sémantických možností.

Modul Paragraphs má v oblasti sémantiky velmi omezené možnosti. Tím, že není možné zasahovat do struktury generovaného HTML, není možné využívat sémantické elementy z HTML5. Zároveň není možné přidávat ani metadatový popis. Tuto možnost neřeší ani žádné existující rozšiřující moduly.















²¹ <https://www.drupal.org/project/paragraphs>

ŠABLONY ČLÁNKŮ *

✚ Type: Základní text

Odstranit

Základní text

B I U S x² x₂        Formát **I_x**        Zdroj

Čtvrté oslavy státního svátku se již zvyklostně ponosou v duchu politických shromáždění, demonstrací a průvodů. Ovšem dvě velké změny, alespoň oproti loňskému roku, nastanou. Albertovu nikdo nemůže upřít spojení s tímto svátkem, totéž platilo i pro studenty, ovšem až do loňského roku, kdy jsme nebyli na Albertov vpuštěni. Letos je vše jinak a právě zde bude probíhat program, jehož obsah spadá pouze pod taktovku vysokoškolských studentů. Na památku bude také Albertov propojen historickou stezkou s Národní třídou. Všichni účastníci tak budou mít možnost projít si celou trasu průvodu ze 17. listopadu 1989, od studentské akce na počátku cesty až ke studentské akci na jejím konci.

Dále by se měly po celé republice v 19:30 (zhruba v tuto dobu byl zahrazen průvod v roce 1989) rozeznít budíky a tím symbolicky probouzet společnost. Budíky budou drnčet především na Václavském náměstí, v rámci akce Koncert pro budoucnost, ovšem organizátoři doufají i v další místa.

✚ Type: Obrázek + popis

Odstranit

Obrázek (1)



Alternative text *

Popis tohoto obrázku















 IMG_20150622_130629_HDR.jpg (2.9 MB)

Odstranit

✚ Type: Základní text

Odstranit

Základní text

B I U S x² x₂        Formát **I_x**        Zdroj

Přehled větších akcí, které můžete navštívit, naleznete níže.

- V 10:00 před Hlavním nádražím proběhne demonstrace Nacionalismus není alternativa pořádaná Iniciativou Ne Rasismu!. Poté pokračuje průvodem přes Opletalovu a Hybernskou ulici a dále Jindřišskou a Vodickovou až na Jungmannovo náměstí, kde má v 16:00 skončit.
- Od 11:00 do 22:00 se na Albertově konají akce organizované Centrem spolků, studentů a absolventů Univerzity Karlovy a studentskou částí Akademického senátu Univerzity Karlovy přímo navazující na pietní akt, který proběhne v 9:00 tradičně u Hlávkovy koleje a v Žitné ulici. Projev zde má mít například rektor Univerzity Karlovy Tomáš Zima, rektor Masarykovy univerzity Mikuláš Bek, či Petr Pithart.
- Od 11:55 se na náměstí Jana Palacha uskuteční Demonstrace na obranu skutečné demokracie, proti nikým nevolným elitám Bloku proti islamizaci, který je jednou z nástupnických organizací rozpuštěného Bloku proti

Vyberte blok obsahu, který chcete vložit

Přidat Základní text

Přidat Obrázek + popis

Přidat Obrázek + popis (vedle sebe)

Přidat Rozhovor

Přidat Citace, otázka apod.

Přidat Odkaz na galerii

Přidat Odkaz na obsah

Přidat Pohled autora

Přidat Otázka + odpověď

Poslat na korekturu

Korektura hotova

Save and keep published

Preview

Smazat

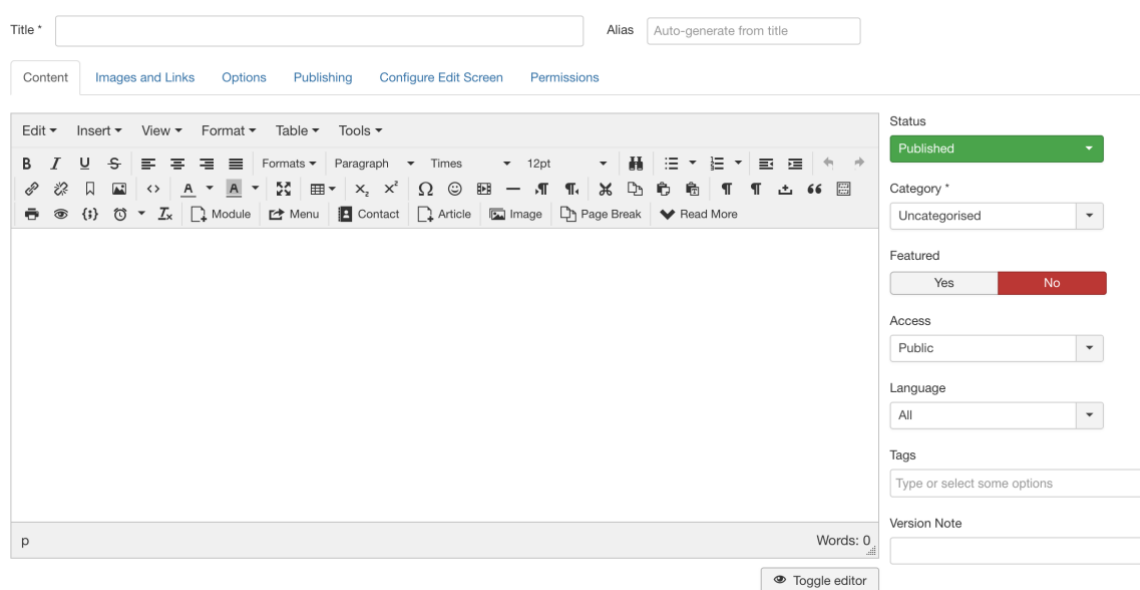
Obrázek č. 17 - Ukázka možností modulu Paragraphs; zdroj: vlastní zpracování

5.3.2 Joomla

V základní instalaci je tento redakční systém nejrozsáhlejší a nabízí nejvíce možností. Pro administraci vyžaduje méně znalostí než systém Drupal a je vhodný pro

webové stránky typu e-commerce, sociální sítě a celkově různé specializované a unikátní obory (Mening, 2015).

Nástroje pro tvorbu obsahu jsou velmi fragmentované a poskytují nejméně možností. V základu nabízí pouze jedno pole s WYSIWYG editorem pro vložení obsahu (viz. Obrázek č. 18). Je možné nadefinovat další pole, ale práce s nimi je omezená. Pomocí těchto polí je možné pouze nadefinovat fixní šablonu a není možné např. měnit pořadí, přidávat další dle potřeby apod.



Obrázek č. 18 - Základní editace obsahu v redakčním systému Joomla; zdroj: vlastní zpracování

Na druhou stranu existuje velké množství rozšíření, která umožňují strukturovat obsah pomocí bloků. Mezi nejvýznamnější patří:

- FLEXIContent,
- Content Builder,
- Form2Content,
- Cobalt,
- Fields and Filters,
- FieldsAttach,
- MiniCCK,

- a další.

Zmíněná rozšíření přistupují ke správě obsahu odlišně a výrazně mění správu obsahu. Zobrazení jednotlivých instancí obsahu a jejich výpisy je nutné řešit pro každý modul odlišně. Tato fragmentace pak ztěžuje vývoj dalších rozšíření např. pro migraci obsahu pro aktualizace, API pro mobilní zařízení apod.

Z hlediska sémantiky a metadatového popisu poskytuje pouze modul FLEXIContent omezené možnosti v podobě přiřazení atributu *itemtype* pro Microdata jednotlivým obsahovým blokům. Uživatelům vytvářejícím obsah je tato možnost skryta.

5.3.3 WordPress

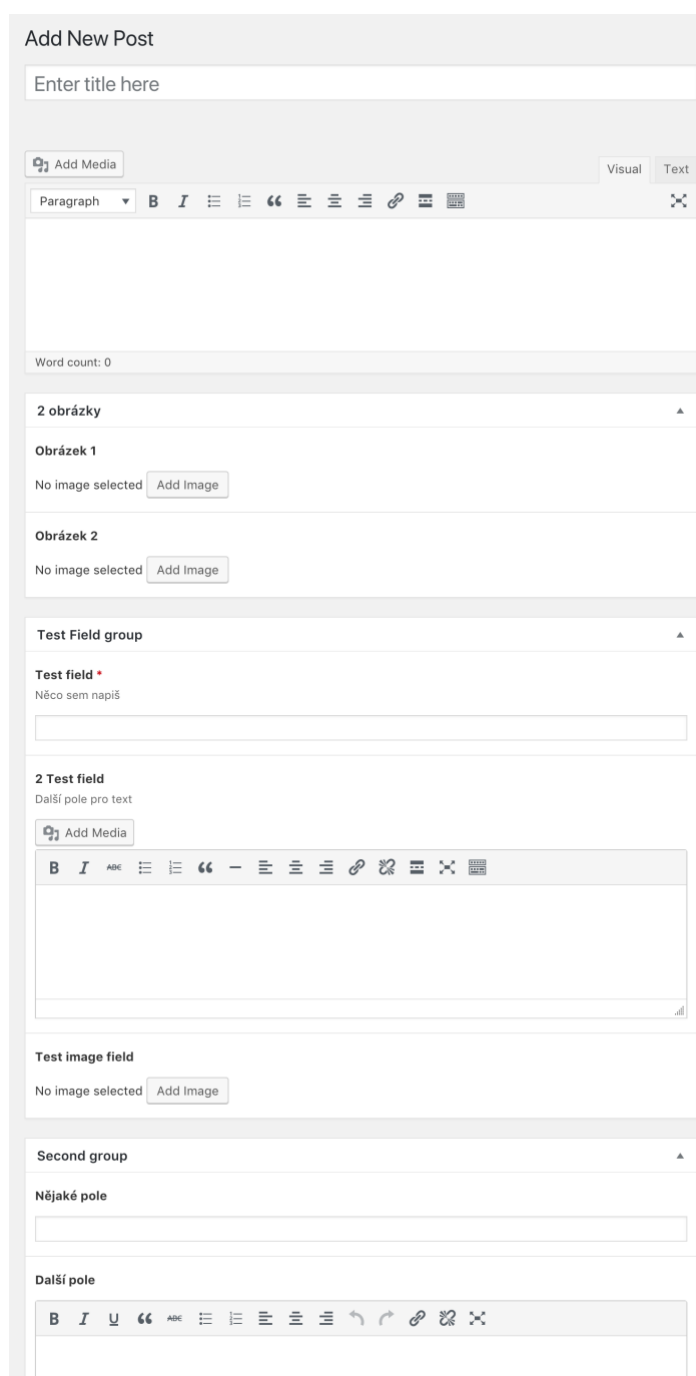
Redakční systém WordPress je ze všech popisovaných nejpobulárnější a vyžaduje nejméně znalostí pro administraci (Mening, 2015). Využívá se často pro správu jednoduchých stránek a dokáží jej ovládat i uživatelé bez odborných znalostí.

Obdobně jako redakční systém Joomla nabízí WordPress pro správu obsahu v základu jedno pole s WYSIWYG editorem. Toto chování lze upravit pouze instalací dodatečných rozšíření. Každý prvek obsahu se nazývá *post*. Toto základní chování lze opět upravit pomocí řady rozšíření.

Nejpoužívanější rozšíření je plug-in Advanced Custom Fields (přes 1 milion stažení)²². To opět umožňuje strukturování obsahu pomocí bloků (polí - fields). Využit lze jak přednastavené typy bloků, tak i vytvářet vlastní (pomocí dalších rozšíření). Celkově je použití tohoto plug-inu nejjednodušší pro nezkušené uživatele vkládající obsah. Na druhou stranu poskytuje méně funkcí. Např. bloky (pole) nelze přidávat k instancím obsahu dynamicky. Náhled editace obsahu s pomocí tohoto pluginu ukazuje Obrázek č. 19. Pokud administrátor definuje pole, jsou u jednotlivých postů přidány automaticky a lze je pouze nevyplnit. Pokud by tedy administrátor přidal příliš mnoho

²² <https://wordpress.org/plugins/advanced-custom-fields/>

polí, uživatelé by tím mohli být zahlceni. Hrozí tedy významné zhoršení použitelnosti. Existuje však i komerční (placená) verze rozšíření, která poskytuje více funkcí.



Obrázek č. 19 - Editace šablony bloků obsahu; zdroj: vlastní zpracování

Pro redakční systém WordPress existuje řada rozšíření, které řeší obohacování obsahu metadaty (např. Schema, WPSSO, WP SEO Structured Data Schema, Schema

App Structured Data). Všechny však pracují na úrovni stránek a jejich konfigurace vyžaduje znalosti problematiky metadat. Žádný nespolupracuje přímo s bloky obsahu z pluginu Advanced Custom Fields.

5.4 Shrnutí

Informační obsah v prostředí WWW dnes již nevkládají pouze programátoři a odborníci se znalostmi webových technologií. Postupně se vyvinuly tzv. WYSIWYG editory, které nabízejí uživatelům podobné prostředí jako klasické desktopové textové procesory. Ukazuje se však, že tyto editory již nestačí a mají řadu nedostatků a v některých případech jsou pro uživatele těžko ovladatelné. Zejména neřeší např. responzivitu nebo sémantiku obsahu. V prostředí WWW se pak pomalu začíná prosazovat přístup v podobě kompozice obsahu z nezávislých bloků a omezování funkcí WYSIWYG editorů.

Pro chod a správu webových stránek se využívají redakční systémy. To jsou kompletní webové aplikace, které zahrnují serverovou i klientskou část. Mezi nejrozšířenější patří Wordpress, Joomla a Drupal. Ty stále technologicky využívají jazyka PHP, primárně využívají databázový server MySQL a jsou open source. Pomalu se ale objevují redakční systémy založené čistě na jazyce JavaScript.

Všechny redakční systémy přistupují ke správě informačního obsahu rozdílně. Je vidět postupný vývoj směrem k pokročilé kompozici obsahu a jeho strukturování. Nejpokročilejší je v tomto redakčním systému Drupal. Ten má podporu strukturování přímo v jádru redakčního systému a další rozšíření pouze přidávají funkce a upravují chování. Je zde také vidět silná dominance jednoho rozšíření (Paragraphs). Naproti tomu redakční systém Joomla nabízí mnoho rozšíření a celkově je díky tomu správa obsahu velmi fragmentovaná. WordPress je v tomto ohledu někde mezi. Nabízí poměrně robustní a dominantní řešení v podobě doplňku ACF. Na druhou stranu nenabízí tolik funkcí a možností přizpůsobení.

K ukládání obsahu všechny redakční systémy používají různé struktury databázových tabulek. Přenositelnost obsahu mezi nimi je závislá na migračních

rozšířeních. Jejich funkce jsou však velmi omezené a jsou schopny přenášet pouze základní obsah, který není strukturován pomocí přidaných rozšíření.

Na základě provedené analýzy je možné identifikovat potenciál pro další výzkum v následujících oblastech.

5.4.1 Vkládání a aktualizace obsahu

Redakční systémy mají v této oblasti řadu nedostatků. Je možné navrhnout některá obecná doporučení pro zlepšení použitelnosti. Vkládání obsahu by mělo být více přizpůsobené méně pokročilým uživatelům, kteří nemají znalosti webových technologií. WYSIWYG editory, ač jsou na první pohled pro uživatele nejjednodušší, přinášejí při editaci strukturovaného obsahu řadu omezení. Nereagují např. Na nutnost responzivity, sémantiky a ze své podstaty k tomu nemají nástroje.

5.4.2 Ukládání obsahu

Současné redakční systémy přistupují celkově rozdílně k ukládání obsahu. Dochází k potížím nejen při migraci obsahu mezi nimi samotnými, dalšími aplikacemi, ale i při aktualizacích na novější verze. Chybí standardizace, která by sjednotila strukturu a pomáhala přenositelnosti. Obsah je také nutné poskytovat v určité formě pro aplikace na mobilních zařízeních. Standardizace by tak pomohla vývoji standardních API a rozhraní pro komunikaci. V této oblasti je tak možné identifikovat široký výzkumný potenciál.

5.4.3 Prezentace, sémantika a archivace

Redakční systémy při správě informačního obsahu nedostatečně řeší problematiku prezentace. Žádné z rozšíření, která poskytují pokročilou kompozici obsahu, neumožňuje změnit strukturu a použité elementy generovaného výstupního HTML. Výstupní vzhled je tak nutné řešit na úrovni obecné šablony pro celý web a jen s pomocí kaskádových stylů. Tento problém také souvisí s responzivitou, která není řešena téměř vůbec. To je vzhledem k počtu přístupů k webovým stránkám

z mobilních zařízení velmi závažný problém – viz. kapitola Media Queries a responzivita webových stránek.

Dále není řešena sémantika na úrovni informačního obsahu. Redakční systémy umožňují popisovat metadaty pouze celé stránky s obsahem. Již ale neumožňují doplňovat metadaty strukturovaný obsah. To je vzhledem k rostoucímu rozvoji umělé inteligence velmi závažný problém – viz. kapitola Sémantika v prostředí WWW.

6 Metodika Web Information Content Management (WICM)

6.1 Hypotéza dizertační práce

Na základě syntézy získaných poznatků z předchozích kapitol bylo zjištěno, že existující metody a nástroje pro vkládání, aktualizaci, uložení a prezentaci obsahu v prostředí WWW jsou nedostatečné a neřeší některé důležité aspekty jako je např. sémantika. Zároveň zde chybí standardizace, která napomáhá vývoji open source softwaru. Dizertační práce se tak bude dále touto problematikou zabývat. Cílem bude navrhnout metodiku pro vkládání, aktualizaci, uložení a prezentaci obsahu v prostředí WWW a tu dále ověřit.

6.2 Cíle a požadavky navrhované metodiky

Hlavním cílem metodiky WICM je zejména zlepšení současného stavu a rozšíření možností práce s informačním obsahem v prostředí WWW. Standardizace v této oblasti usnadňuje vývoj softwaru, umožňuje vývoj široce použitelných knihoven, modulů a frameworků založených na otevřených zdrojových kódech (Sahay, 2003). Jak ukazuje Merilina a Matinlassi (2006) použití open source softwarových komponent má pozitivní vliv především na rychlost vývoje software. Snazší a rychlejší přechod mezi redakčními systémy, nebo přenos obsahu tak může mít značný ekonomický přínos pro rozvoj firem a organizací.

Návrh metodiky reaguje na identifikované nedostatky současného stavu vymezené v předchozích kapitolách. Samotná metodika se skládá ze tří dílčích částí:

- vkládání a aktualizace obsahu,
- ukládání obsahu,
- prezentace, sémantika a archivace.

Metodika by měla vycházet ze současných technologických možností, existujících standardů a specifikací, pragmaticky přistupovat k novým možnostem a technologiím, které mohou přijít nebo jsou již ve fázi vývoje. Na druhé straně by měla být platformě nezávislá a neměla by být vázána na konkrétní implementace, programovací jazyky, frameworky, či databázové systémy.

6.2.1 Zaměření metodiky

Metodika je tematicky zaměřena na informační obsah v prostředí WWW. Ten je vzhledem k zaměření práce vymezen v kapitole 4.3. V praxi se jedná o obsah, který může být reprezentován elementem *article* v rámci specifikace HTML5. Jde tedy především o články online médií. Metodika by měla být využitelná především vývojáři, kteří vyvíjí nebo používají open source redakční systémy a jim podobné aplikace, ale také uživateli, kteří vytvářejí a vkládají informační obsah na webové stránky.

6.2.2 Identifikace výchozích požadavků a nedostatků současného stavu

Vkládání a aktualizace obsahu

Informační obsah v prostředí www by měli být schopni vkládat a aktualizovat i uživatelé bez znalostí technologií HTML a CSS (Brown, 2014). V současné době se většinou pro vkládání a aktualizaci obsahu používají tzv. WYSIWYG editory. Jejich podstatou je, že autor přímo edituje obsah tak, jak jej ve výsledku uvidí uživatel (čtenář). To se postupem času ukazuje jako nepřiliš vhodné řešení. Vznikají problémy s nekonzistentností výsledného HTML kódu (Spiesser a Kitchen, 2004), autoři mívají často problémy s některými složitějšími částmi obsahu (Masner et al., 2015), WYSIWYG editory mohou být hrozbou pro bezpečnost (Javed a Schwenk, 2015) a dle Khalili et al. (2012) a Khalili a Auer (2015) nepodporují sémantiku a popis metadaty.

Syntézou poznatků z předcházejících kapitol a provedených analýz lze definovat základní požadavky následovně:

- Vkládat a aktualizovat obsah by měli být schopni i uživatelé, bez znalostí technologií jako je HTML, CSS a JavaScript.

- Vkládání a aktualizace obsahu musí umožňovat minimalizaci chyb na vstupu, které by mohly vést k nekonzistentnímu výstupu (z hlediska HTML).
- Vstup dat by měl být pro uživatele co nejjednodušší.
- Uživateli by mělo být zobrazeno, jakým způsobem bude vypadat výstup vkládaných dat.

Ukládání obsahu

Většina redakčních systémů je v současné době postavena na technologiích PHP, MySQL. K ukládání obsahu se tak využívá relačních databází. Každý redakční systém však přistupuje k ukládání i vkládání obsahu rozdílně. V současné době je však možné pozorovat nové přístupy k vývoji redakčních systémů za použití technologií Node.js, JavaScript a NoSQL databází. Takto postavené systémy prozatím nedosahují takové rozšířenosti ani nabídky funkcí. K ukládání obsahu však opět přistupují rozdílně. Do budoucna je ale možné očekávat širší rozšiřování. Jedním z problémů, který může nastat, je pak přenos obsahu mezi těmito systémy. Na jedné straně jsou data ukládána do SQL databází, na straně druhé může být struktura dat v NoSQL databázích značně odlišná.

Standardizace v oblasti ICT značně pomáhá rozvoji a rozšiřování technologií, ale i samotných aplikací (Canazza, 2009; Selwyn, 2011). Tato tendence je zřejmá i v případě samotného HTML, které je téměř po celou svou historii spravováno konsorciem W3C. Standardizace struktury ukládaného obsahu nad úrovní konkrétních databázových systémů tak může přinést řadu výhod. Může usnadnit přechod mezi redakčními systémy, aktualizaci, sdílení, ale například i vývoj různých softwarových nástrojů, modulů, frameworků a knihoven.

Syntézou poznatků a analýz z předcházejícího výzkumu lze definovat základní požadavky následovně:

- Metodika by měla uvést obecný model ukládaných dat, který bude nezávislý na použité databázové technologii.

- Struktura ukládání by měla odpovídat požadavkům pro vkládání a aktualizaci obsahu a zároveň obecně přijímaným požadavkům na obsah v prostředí WWW (sémantika, základní přístupnost, popis metadaty apod.).
- Model ukládaných dat by měl být schopen pojmout téměř jakýkoli informační obsah, který je technologicky možný publikovat v prostředí WWW.
- Model ukládaných dat by měl být rozšiřitelný.

Prezentace, sémantika a archivace

Výsledná prezentace obsahu je z velké části závislá na aplikaci, která jej zobrazuje uživatelům. V prostředí WWW je to především ovlivněno kaskádovými styly (CSS) a také strukturou HTML, ve které je obsah uchován a zobrazen ve webovém prohlížeči. Webové stránky v průběhu času mění svůj vzhled. Kromě změny technologie redakčního systému nebo aplikace, která obsah zobrazuje, často dochází k redesignu tak, aby odpovídaly moderním trendům. Pro prezentaci dat je tak možné definovat několik výchozích požadavků:

- Ukládaný obsah by neměl obsahovat CSS kód.
- Mělo by být možné definovat základní pravidla zobrazení, zejména strukturu HTML s ohledem na snadnou úpravu při změně vzhledu.
- Struktura uloženého obsahu by měla být snadno transformovatelná pro zobrazení obsahu v aplikacích pro mobilní zařízení.

Obsah v prostředí WWW není prezentován pouze uživatelům – lidem, ale také počítačům a jejich programům. Jedná se např. o indexovací nástroje vyhledávačů, chytré asistenty, metadatové harvestery nebo sociální sítě a služby pro sdílení obsahu. Základním aspektem je zde dobře strukturovaný obsah, který odpovídá základním sémantickým pravidlům. Výchozí požadavky jsou tak následující:

- Metodika musí respektovat základní sémantická a metadatová pravidla.
- Model uložených dat musí být rozšiřitelný o sémantický metadatový popis.

- Metodika musí respektovat základní pravidla pro optimalizaci pro vyhledávače (SEO).

Webové stránky a aplikace v průběhu času mění technologie nebo redakční systém, na kterém jsou postaveny. Může se jednat o aktualizaci redakčního systému na novou verzi, nebo přechod na úplně jiný. Různé redakční systémy v současné době pracují s obsahem odlišně a při přechodu na jiný redakční systém je tak nutné transformovat obsah dle aktuálních potřeb. To je často velmi náročné a takový přechod může způsobit i mnoho problémů a ekonomických nákladů, jelikož struktura ukládaného obsahu může být velmi odlišná. Z těchto faktů vycházejí následující základní požadavky:

- Musí být stanoven jednotný formát pro přenos obsahu, nezávislý na implementační aplikaci.
- Tento formát musí umožnit i archivaci obsahu.

6.3 Návrh metodiky

6.3.1 Výchozí prototyp

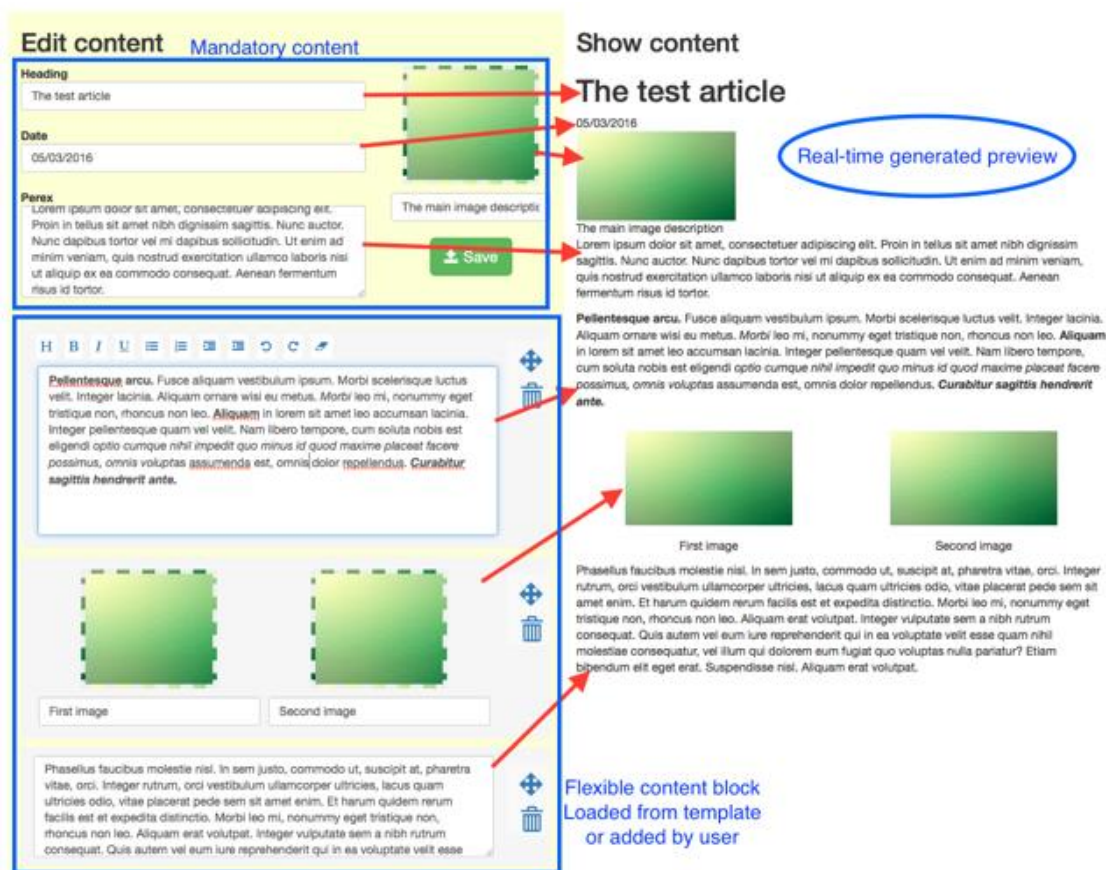
Návrh výchozího prototypu byl publikován v „Prototype of a Content Creation and Updating Application Module for Agrarian Sector and Regional Development“ (Masner et al., 2016). Pro vývoj byl zvolen framework Ember. Ten společně s Ember-CLI umožňuje velmi rychlý vývoj s dostatečnou mírou abstrakce. To je pro vývoj prototypů optimální. Framework je založen na klasickém návrhovém vzoru MVC (Model View Controller). Pro zjednodušení vývoje nezbytné serverové části prototypu bylo zvoleno řešení Google Firebase. To v sobě především zahrnuje NoSQL databázi, která je založena na ukládání struktur podobných těm, které používá formát JSON. Zároveň poskytuje knihovny pro Ember.

Vkládání a aktualizace obsahu

Vkládání a aktualizace obsahu je řešena pomocí kompozice dílčích bloků. Tento koncept používá modul Paragraphs pro redakční systém Drupal (Nikolić a Šilc, 2016).

Obsah je tak tvořen z povinné části, která obsahuje titulek článku, datum, úvodní odstavec a nepovinně také úvodní obrázek, a dynamických bloků. Tyto bloky nejsou statické, ale jejich pořadí se může měnit tak, jak tvůrce obsahu potřebuje.

Koncept WYSIWYG editace je v prototypu zpracován odlišným způsobem, než se běžně používá. Na rozdíl od zobrazení výsledku přímo v editační části, je výsledek zobrazen samostatně. Autorovi je tak zobrazen skutečný formát, v jakém bude obsah zobrazen. Jde tedy o faktické oddělení částí WYS a WYG. U klasických WYSIWYG editorů často dochází ke zkreslení skutečného výsledku. Tento nově navržený koncept je znázorněn na následujícím obrázku.



Obrázek č. 20 - Základní prototyp a princip oddělení WYS (What You See) a WYG (What You Get); zdroj: (Masner et al., 2016)

Ukládání obsahu

Ukládání obsahu vychází primárně z tvorby a vkládání. Musí být umožněno jej zpětně editovat a není tak možné celý obsah převést do formátu HTML. Ukládány jsou jednotlivé bloky samostatně. Obsah je tak složen ze dvou hlavních částí:

1. Povinné položky (titulek, datum, úvodní odstavec apod.);
2. Hlavní obsah sestavený z bloků.

Uložení je řešeno v NoSQL databázi Firebase, která ukládá data do struktur podobných formátu JSON. Záznam jedné instance obsahu je pro ilustraci zobrazen na následujícím obrázku.



Obrázek č. 21 – Struktura dat výchozího prototypu v NoSQL databázi Firebase; zdroj: vlastní zpracování

Obecně je struktura ukládaného obsahu znázorněna v následující kapitole na diagramu č. 1. V dalších částech práce pak bude struktura ukládání podrobena bližšímu zkoumání a upravena.

Prezentace, sémantika a archivace

Oblast prezentace dat je ve výchozím prototypu řešena tak, že pro každý blok obsahu je definována tzv. komponenta z frameworku Ember. Ta nese strukturu ve formátu HTML, která určuje, jak se jednotlivá data z bloku zobrazí v prohlížeči. K tomu je definován základní předpis kaskádových stylů.

Vzhledem k architektuře frontendového frameworku Ember, který pracuje na straně klienta přímo s modely a na jejich úrovni komunikuje s backendem, formát přenosu obsahu mezi klientem a serverem nebylo nutné řešit. Tato komunikace probíhá na úrovni dat a transformace obsahu pro prezentaci je tak řešena přímo v aplikaci na straně klienta.

Oblasti sdílení, archivace ani přenositelnost nebyly v rámci výchozího prototypu nijak řešeny.

6.3.2 Výchozí formulace metodiky

Vkládání a aktualizace obsahu

V této části se metodika WICM skládá z doporučení, která by měla splňovat aplikace, která bude metodu implementovat. Tato doporučení jsou následující:

1. Vkládat a aktualizovat obsah by měli být schopni i uživatelé bez znalostí technologií HTML, CSS a JavaScript.
2. Práce s obsahem by neměla být řešená formou jednoho komplexního WYSIWYG editoru. Obsah by měl být skládán strukturovaně z dynamických bloků.
3. Základní principy práce s textem by měly vycházet z toho, na co jsou uživatelé zvyklí (např. z textových procesorů jako je Microsoft Word).

4. Tyto bloky by neměly mít fixní pozici a autor by měl mít možnost s nimi manipulovat a přesouvat je mezi sebou.
5. Autor obsahu by měl mít okamžitou možnost vidět, jak bude vypadat výsledný obsah zobrazený čtenářům.

Uložení obsahu

Metodika WICM pro oblast ukládání dat definuje strukturu ukládaného obsahu formou Doménového modelu v rámci Class diagramu dle notace UML a doplňujících pravidel. Tato struktura odpovídá navrženému prototypu. Její definici znázorňuje Diagram č. 1.

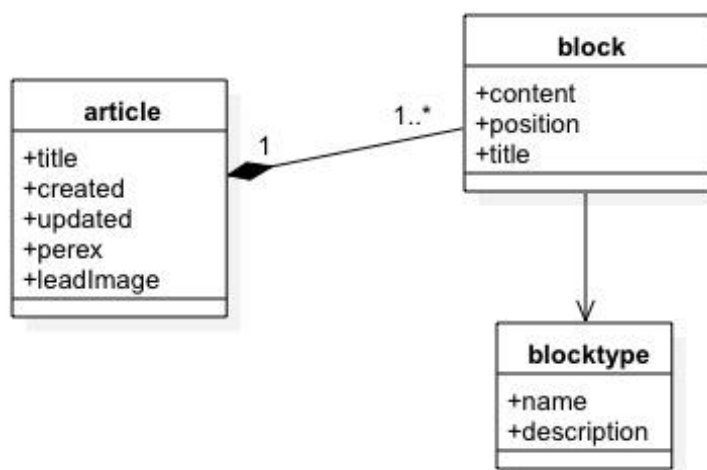


Diagram č. 1 Základní struktura obsahu – Class diagram UML, doménový model; zdroj: vlastní zpracování

6.3.3 Vkládání a aktualizace obsahu

Potřeby uživatelů

Pro bližší zjištění potřeb a požadavků uživatelů a autorů obsahu byla použita metoda Focus Groups. Dle této metody bylo realizováno sezení s vybranými uživateli (redaktory a autory obsahu). Během sezení byl představen výchozí prototyp a následně veden rozhovor ohledně používání současného systému i potřeb

a požadavků uživatelů. Pro realizaci sezení byl stanoven předpis několika základních otázek. Na jejich základě byl veden rozhovor. Otázky byly následující:

1. Používáte WYSIWYG editor a vyhovuje vám?
2. Jaký je postup při tvorbě obsahu? Využíváte MS Word, nebo píšete obsah přímo v redakčním systému?
3. Jak je strukturován obsah, který běžně vytváříte?
4. Co byste řekli na to, kdybyste skládali obsah z oddělených bloků?
5. Jak by vypadala ideální aplikace pro vkládání obsahu?
6. Jak hodnotíte představený prototyp? Co se vám na něm líbí a co byste naopak změnili?

V rámci aplikace metody Focus Groups byly sumarizovány nejdůležitější poznatky od uživatelů. Jak upozorňuje Nielsen (1997) metoda může produkovat nepřesná data, jelikož uživatelé si mohou myslet, že chtějí jinou věc, než doopravdy potřebují. Je tak nutné zjištěné požadavky podrobit kritické analýze a zhodnotit jejich důležitost a proveditelnost. Nejrelevantnější zjištěné poznatky a požadavky jsou následující:

- Uživatelé nejčastěji vytváří obsah v textovém procesoru MS Word a podobných. Následně jej kopírují do redakčního systému.
- Účastníci hodnotili kladně, skládání pomocí bloků, zejména pro komplexnější části obsahu, se kterými se ve WYSIWYG editoru pracuje obtížně.
- Obsah by měl být v reálném čase ukládán pro možnost ztráty připojení k internetu, napájení počítače apod.
- Byly diskutovány některé základní struktury obsahu, avšak ty jsou závislé na webové stránce, kde vycházejí. Existují však některé druhy obsahu, které by měly mít opakující se strukturu.
- Systém pro vkládání obsahu by tak měl umožňovat definici šablon pro některé standardní druhy obsahu – tak jak bylo definováno v předchozím bodě.

- Šablony by neměly být statické. V průběhu času by je mělo být možné měnit. Zároveň by mělo být možné měnit strukturu pro každý obsah (článek) zvlášť, zejména pro zkušenější či odborné uživatele.
- Uživatelé (alespoň někteří – např. editoři, šéfredaktoři) by měli mít možnost tyto šablony vytvářet a upravovat.
- Účastníci hodnotili kladně možnost okamžitého náhledu výsledného zobrazení, ale uvítali by možnost náhledu zobrazení na tabletu, či mobilním telefonu.
- V představeném prototypu chybí některé důležité prvky, které se k obsahu připojují (např. autoři článku, fotogalerie, kategorizace článků apod.)

Na základě nově stanovených požadavků byl aktualizován vyvíjený prototyp. Byla přidána např. funkce ukládání v reálném čase. Obsah může být vytvářen pomocí šablon, které sdružují jednotlivé bloky obsahu do skupin. Tato funkce bude dále podrobena bližšímu zkoumání z hlediska ukládání obsahu a prezentace.

6.3.4 Ukládání obsahu

Atomizace obsahu

Nejprve je nutné provést analýzu toho, co je možné v prostředí WWW v rámci informačního obsahu ukládat a zobrazovat. Obsah bude atomizován na nejmenší použitelné jednotky, které je možné vkládat, ukládat a zobrazovat ve webových prohlížečích. Z již provedeného výzkumu vyplývá, že jazyk HTML je možné použít pro samotné uchování obsahu, ale není vhodný pro samotné strukturování (Masner et al., 2015).

Specifikace HTML5 definuje ke každému elementu žádnou, nebo více kategorií, do kterých spadá. Dále tzv. Content model, tj. obsah (objekty), které může obsahovat. Více je o tomto pojednáno v části 4.4 Webové technologie.

V následujícím textu jsou uvedeny jednotlivé HTML elementy, které mohou nést atomizovaný obsah. Takto atomizovaný obsah je možné ukládat do samostatných

bloků a vytvářet pro ně vstupní formulářová pole. Z těch je následně možné vytvářet složitější struktury, nebo šablony.

- **Nadpisy <hX>**
 - Pro nadpisy existuje v HTML5 celkem 6 elementů – h1, h2, h3, h4, h5, h6. Číselné označení značí úroveň nadpisu. Uvnitř elementu mohou být jakékoliv elementy spadající do kategorie *Phrasing content*²³.
- **Odstavce <p>**
 - Odstavce jsou základním nositelem informačního obsahu a formátovacím elementem v HTML. Obsah v nich lze dále formátovat pomocí elementů spadajících do ckece *Phrasing content*.
- **Seznamy , , <dl>**
 - Ve specifikaci HTML 5 existují dva základní druhy seznamů. Jedním jsou jednoduché seznamy položek (ul, ol), druhým pak seznam definic (dl).
 - Seznamy ul a ol mohou obsahovat pouze položky li, případně tzv. *script-supporting*²⁴ elementy. Jednotlivé položky seznamu pak mohou obsahovat elementy z kategorie *Flow content*²⁵, tedy opět i další seznam.
 - Seznam definic dl pak může obsahovat skupiny položek skládajících se z jednoho a více elementů dt a jednoho nebo více elementů dd, případně tzv. *script-supporting* elementy. Jednotlivé položky dt pak mohou obsahovat elementy z kategorie *Flow content* (s výjimkou elementů *header*, *footer*, kategorie *Sectioning content*²⁶ a *Heading content*²⁷).

²³ <https://www.w3.org/TR/html5/dom.html#phrasing-content-1>

²⁴ <https://www.w3.org/TR/html5/dom.html#script-supporting-elements-0>

²⁵ <https://www.w3.org/TR/html5/dom.html#flow-content-1>

²⁶ <https://www.w3.org/TR/html5/dom.html#sectioning-content-0>

²⁷ <https://www.w3.org/TR/html5/dom.html#heading-content-0>

- **Tabulky <table>**

- Tabulky se skládají z řádků (<tr>) a jednotlivých buněk (<td>). Atomickým prvkem by tak bylo buňka tabulky, avšak situace je obdobná jako u výše zmíněných seznamů. Buňka tabulky musí být vždy v řádku a ten uvnitř tabulky. Tabulky lze ještě členit do sémantických sekcí doplněním některých dalších elementů, např. <thead><tbody> a <tfooter> nad úroveň řádků. Textový obsah buněk lze opět dále formátovat pomocí elementů z kategorie Flow content.

- **Vložený obsah (Embedded content)**

- Dalšími elementy, které mohou nést atomický obsah jsou elementy z kategorie Embedded content²⁸. Jedná se o elementy <audio>, <canvas>, <embed>, <iframe>, , <math>, <object>, <svg> a <video>. Všechny tyto elementy nemají žádný obsah. Jejich zobrazení je určeno hodnotami jejich atributů.

Jak je vidět, specifikace HTML5 je velmi benevolentní, co se týče zanořování jednotlivých elementů. Pro potřeby vytvářené metodiky a atomizaci obsahu je nutné zvolit určitá omezení a pravidla při ukládání informačního obsahu. obsahuje nová pravidla pro *Content model* jednotlivých elementů. Omezení definovaná ve specifikaci HTML5 však mají stále přednost. Toto omezení platí pro základní úroveň ukládaného obsahu, tedy všechny třídy, které jsou v rámci diagramu č. 2 navázány na třídu Block vazbou generalizace. Další zanoření, které respektuje pravidla HTML5 je možné vytvářet pomocí skupin bloků (třída Block group). Takto definovaná pravidla následně umožní jednoznačně ukládat strukturovaný obsah. Z něj je poté možné jednoduše a automatizovaně obsah zobrazovat a vykreslovat.

²⁸ <https://www.w3.org/TR/html5/dom.html#embedded-content>

Element	Nový content model
h1, h2, h3, h4, h5, h6	Elementy ze sekce 4.5 Text-level semantics ²⁹
p	Elementy ze sekce 4.5 Text-level semantics
ul, ol	Pouze elementy li
dl	Pouze elementy dt a dd
li, dd, dt	Elementy ze sekce 4.5 Text-level semantics
table	caption, thead, tbody, tfoot, tr
td	Elementy ze sekce 4.5 Text-level semantics Elementy z kategorie embedded content

Tabulka č. 2 Pravidla pro nový content model na nejnižší úrovni ukládání obsahu

Z části 6.3.3 vyplývá, že uživatelé potřebují co nejjednodušší rozhraní pro vkládání, které je zároveň povede a nebudou se muset zabývat strukturováním jednotlivých částí obsahu pro finální prezentaci. Z toho důvodu je nutné podporovat vytváření složitějších bloků obsahu a celých šablon pro konkrétní typizované články. Toto je v diagramu č. 2 řešeno pomocí třídy Block group. Ta je záměrně navázána vazbou agregace sama na sebe tak, aby bylo možné tvořit libovolné skupiny a zanořovat je do sebe. Tím je možné vytvářet libovolné šablony. V následujících kapitolách bude blíže pojednáno o položce *composition rule*, která bude určovat, jak budou jednotlivé blok složeny a zobrazeny.

Definice modelu uložení

Syntézou poznatků z předchozích kapitol, zejména atomizace obsahu a části vkládání a aktualizace obsahu byl navržen model pro uložení dat, který by měla

²⁹ <https://www.w3.org/TR/html5/text-level-semantics.html>

aplikace podle metodiky WICM implementovat. Model byl navržen pomocí notace UML.

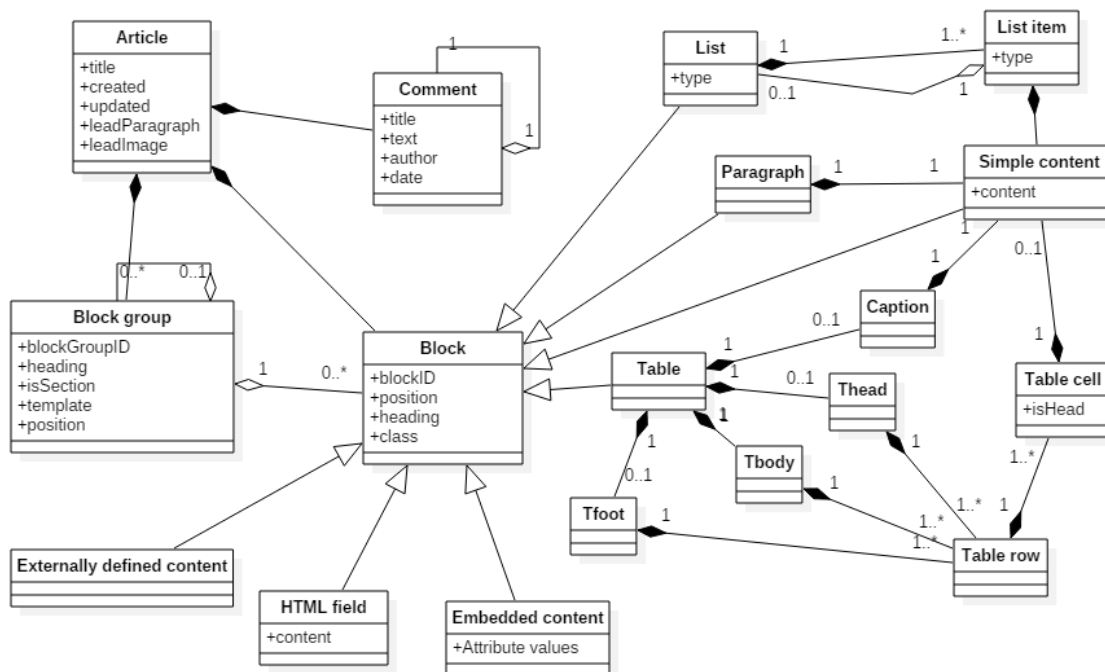


Diagram č. 2 Model ukládaného obsahu - Class diagram UML, doménový model; zdroj: vlastní zpracování

Article

Třída *Article* nese hlavní a obecné informace o obsahu (např. článku). Zde jsou uloženy informace, jako je nadpis, datum publikace apod. Na ní jsou navázány jednotlivé bloky obsahu (*Block*) a skupiny bloků (*Block group*).

Block

Blok je základní stavební prvek strukturovaného obsahu. Obsahuje atomizovanou část obsahu.

Block group

Jednotlivé bloky mohou být pro složitější obsahové prvky slučovány do skupin. Pro tyto skupiny je pak možné vytvářet šablony. Skupiny bloků mohou poté volitelně

obsahovat další skupiny. Každá skupina by pak měla nést pravidlo, jak jsou její jednotlivé bloky prezentovány. To je řešeno pomocí atributu *template*, který obsahuje HTML šablonu pro zobrazení v prohlížeči.

List, Paragraph, Table, Embedded content, Simple content

Tyto třídy nesou samotný atomizovaný obsah dle jeho typu. Obsah ve formátu HTML je omezen definicí nového Content modelu, který ukazuje Tabulka č. 2. Tento obsah by již nadále neměl být dělen a tvoří základ struktury. Ve třídě Simple content pak nesmí být jiný obsah, než ten, který je definován sekci 4.5 Text-level semantics specifikace HTML 5.

Externally defined content

Redakční systémy často připojují část obsahu z jiného zdroje. To může představovat např. blok s podobnými články, fotogalerii, část textu z jiného článku, anketu, formulář apod. Tento obsah může být často interaktivní a jeho konkrétní podoba může se měnit. O finální prezentaci se tak stará samotný redakční systém, nebo aplikace, která obsah generuje.

HTML field

Ačkoli by mělo být možné veškerý informační obsah začlenit do tříd zmíněných výše, zkušenější uživatelé by měli mít možnost vložení i nestandardního obsahu ve formátu HTML. Ten může představovat např. interaktivní kód v jazyce JavaScript, reklamní bloky apod.

6.3.5 Prezentace, sémantika a archivace

Prezentace uživatelům

Obsah v prostředí WWW je primárně určen pro uživatele, tedy lidi. V literatuře jsou často označováni jako koncoví uživatelé (end users). Pro ty je nejdůležitější vzhled a vizuální struktura obsahu. To se na webu řeší pomocí vhodné struktury HTML ve spojení s kaskádovými styly (CSS). Kaskádové styly jsou vždy vázány na konkrétní

implementaci aplikace nebo redakčního systému. Zároveň se v průběhu jeho životního cyklu mohou měnit. Webové stránky často mění vzhled tak, aby odpovídaly aktuálním trendům nebo potřebám. Je však nutné stanovit základní pravidla pro strukturu HTML. Zejména se jedná o sestavení skupin bloků (šablon) a bloků samotných.

Pravidla zobrazení

Následující popis vychází z Diagramu č. 2.

1. Každý prvek z třídy Block je obalen elementem, který vychází z konkrétní třídy:
 - a. List: , , nebo <dl>
 - b. Paragraph: <p>
 - c. Simple content: <div>
 - d. Table: <table>
 - e. Embedded content: <audio>, <canvas>, <embed>, <iframe>, , <math>, <object>, <svg>, <video>
 - f. HTML field: <div>
 - g. Externally defined content: <div>
2. K tomuto elementu je přidán HTML atribut class s obsahem z UML atributu class třídy Block.
3. Každý prvek třídy Block group sestává z prvků třídy Block, případně i Block group. Sestavení je určeno šablonou HTML (atribut composition rule). Přesná specifikace této šablony je uvedena ve finální formulaci metodiky.
4. Každá instance obsahu (třída Article) je sestavena z prvků tříd Block a Block group. Řazení vychází z atributu position. Obsah je obalen elementem <article>. Základní informace, které nese třída Article by měli obaleny v elementu <header>. Samotné sestavení je opět určeno šablonou HTML (atribut composition rule). Přesná specifikace této šablony je uvedena ve finální formulaci metodiky.

Sémantika

Kromě uživatelů mohou k obsahu přistupovat i aplikace a služby. To mohou být např. indexovací nástroje vyhledávačů, služby pro agregování obsahu, čtení metadat, sdílení obsahu, umělá inteligence, aplikace strojového učení apod. Aby obsahu rozuměly i aplikace a stroje obecně, musí být obsah v první řadě strojově čitelný (machine-readable). Dalším krokem je, aby obsahu počítače i rozuměly. Měl by tedy být správně strukturován a popsán metadaty.

Základem je tedy správná struktura výsledného HTML kódu. Na základě specifikace HTML5 a navrženého modelu je pak možné definovat minimální šablonu a elementy, které by měla každá jednotka obsahu obsahovat. Zdrojový kód č. 2 Základní HTML šablona s povinnými elementy znázorňuje základní šablona s elementy, které by měly být vždy přítomné.

```
<article>
  <header>
    <h1>Title of the content</h1>
    <time datetime="2016-02-14 20:00">14. 2. 2016 20:00</time>
    
    <p>
      Lead paragraph...
    </p>
  </header>
  <section>
    All blocks and block groups
  </section>
  <footer>
    Author information etc.
  </footer>
</article>
```

Zdrojový kód č. 2 Základní HTML šablona s povinnými elementy

Celý obsah by měl vždy být uzavřen v elementu `<article>`. Ten může být vynechán pouze v případě, že tvoří jediný obsah na stránce. Další povinné elementy pak jsou:

- `<header>` (základní informace o obsahu)
 - `<hX>`
 - Nadpis celého obsahu.

- Úroveň nadpisu vychází vždy ze struktury stránky, kde je článek vkládán.
- Vždy jen jednou, podnadpisy by dle (W3C, 2014a) neměli být v elementu h.
- <time>
 - Datum publikování obsahu.
 - Může být doplněno o další elementy time, např datum úpravy apod.
- <p>
 - Úvodní odstavec (perex).
- <section>
 - Tělo obsahu.
 - Obsahuje vše ze tříd Block a Block group definovaných v kapitole 6.3.4.
- <footer>
 - Informace o autorovi a další doplňující informace.

Dalším krokem je popis obsahu metadaty. K tomu se dá využít několika formátů zápisu. Nejrozšířenější jsou Microdata, RDFa a JSON-LD. Všechny tři je nutné využít ve spojení s nějakým slovníkem, který určuje význam. V prostředí WWW jsou nejrozšířenější slovníky iniciativy Schema.org. Každý ze zmíněných zápisů je vhodný k jinému účelu. Pro potřeby navrhované metodiky je tak vhodnější popis metadaty na základě slovníku. Jednotlivé formáty je pak možné generovat pomocí softwarových knihoven dle potřeby.

Pro obecný popis celého obsahu je možné využít např. typ *Article*³⁰ nebo z něj odvozený specifitější (*NewsArticle*, *Report*, *ScholarlyArticle*, *SocialMediaPosting* nebo *TechArticle*). Jednotlivé třídy *Block* a *Block group* pak mohou nést data o samostatných entitách. Diagram č. 3 znázorňuje aktualizovaný model uložení, který umožňuje

³⁰ <http://schema.org/Article>

ukládat metadata. Názvy atributů byly upraveny tak, aby odpovídali přímo vlastnostem ze slovníku *schema.org*.

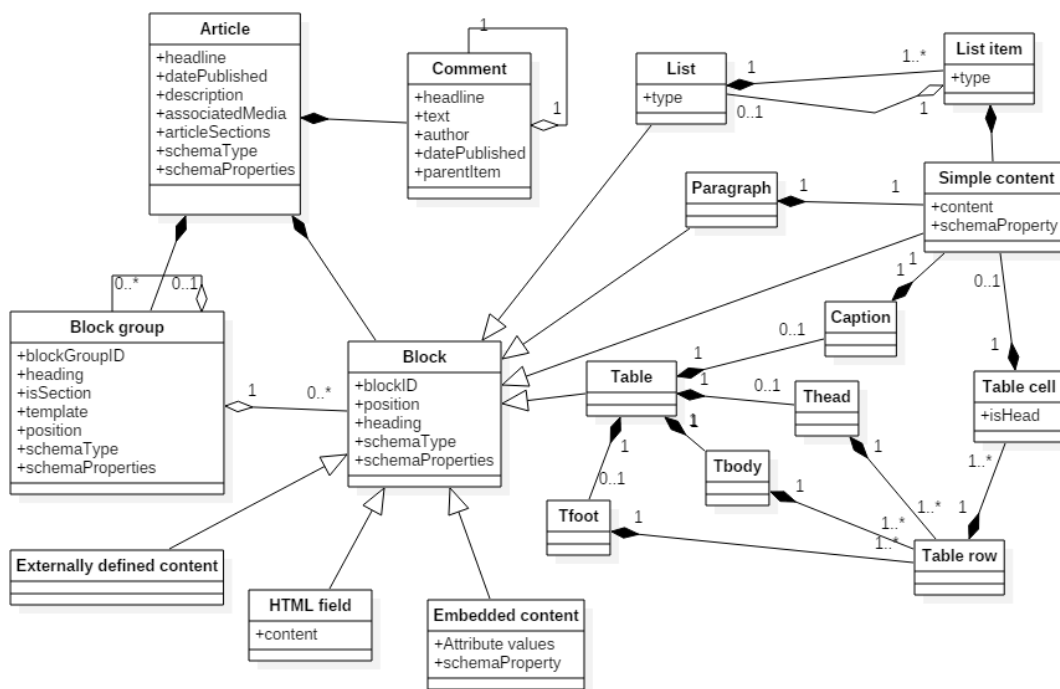


Diagram č. 3 - Model uložení obsahu doplněný o podporu metadatového popisu; zdroj: vlastní zpracování

Vysvětlení přidáných atributů:

schemaType

- Představuje typ (název) použitého schématu³¹.
- Pro Microdata se jedná o atribut *itemtype*, v rámci JSON-LD je to objekt *@context*, v RDFa pak atribut *typeof*.

³¹ <http://schema.org/docs/schemas.html>

schemaProperties

- Atribut schemaProperties je asociativní pole jednotlivých vlastností (Property) z příslušného slovníku. Tyto vlastnosti jsou doplňující a nemusejí být zobrazeny uživateli.

schemaProperty

- Tento atribut představuje název vlastnosti. Její hodnota je pak uvnitř bloku a je obsahem příslušného HTML elementu.

Základní HTML šablona, kterou ukazuje Zdrojový kód č. 2, může být následně upravena s využitím syntaxe pro Microdata takto:

```
<article itemscope itemtype="http://schema.org/Article">
  <header>
    <h1 itemprop="headline">Title of the content</h1>
    <time itemprop="datePublished" datetime="2016-02-14 20:00">14. 2. 2016 20:00</time>
    <div itemprop="associatedMedia">
      <span itemscope itemtype="http://schema.org/ImageObject">
        <img itemprop="contentURL" src="" alt="...">
      </span>
    </div>
    <p itemprop="description">
      Lead paragraph...
    </p>
  </header>
  <section itemprop="articleBody">
    All blocks and block groups
  </section>
  <footer>
    <div itemprop="author" itemscope itemtype="http://schema.org/Person">
      <span itemprop="givenName">Author</span> <span itemprop="familyName">Name</span>
    </div>
  </footer>
</article>
```

Zdrojový kód č. 3 - Základní HTML šablona doplněná o metadata ve formátu Microdata **Error! Reference source not found.**

Archivace a přenos obsahu

Pro možnost snadné archivace a přenosu dat je nutné stanovit standardizovaný formát. K těmto účelům slouží především značkovací jazyky. V prostředí WWW se nejvíce využívají jazyky XML a JSON. V poslední době se spíše využívá druhý jmenovaný. Ten je méně datově náročný a snadněji programově zpracovatelný, díky

své blízkosti ke strukturám známým z programovacích jazyků. Jeho využití v komunikaci s mobilními zařízeními a zpracování je rychlejší než v případě XML (Nurseitov et al., 2009; Masner et al., 2014). Formát JSON lze také přímo, nebo s minimálními úpravami použít v některých NoSQL databázích.

Pro standardizovanou definici datové struktury ve formátu JSON v současné době existuje pouze schematický jazyk JSON Schema (K. Zyp a G. Court, 2013). Pro účely návrhu metodiky bude tedy použito aktuálně nejnovější verze *draft-wright-json-schema-01* dle (Wright, 2017). Ve finální formulaci metodiky tak bude uveden archivační formát pomocí této verze. Tento archivační formát je pak možné použít i pro přenos mezi klientskou částí aplikace (mobilní zařízení, webový prohlížeč) a serverovou např. v rámci json:api³², nebo při exportu obsahu pro různé účely (změna aplikace apod.). Ukázka zastřešujícího schématu je uvedena ve finální verzi metodiky (Zdrojový kód č. 4).

6.4 Formulace metodiky WICM

6.4.1 Konvence

Metodika WICM je formulována obdobně, jako současné webové standardy a specifikace. V textu jsou obsažena klíčová slova indikující úroveň požadavků, která odpovídají anglickým ekvivalentům uvedeným v doporučení IETF RFC2119³³. Přesný ekvivalent a použitá klíčová slova jsou uvedeny v následující tabulce.

Klíčové slovo dle RFC2119	Český ekvivalent
MUST	muset
MUST NOT	nesmět
SHOULD	měl by

³² <http://jsonapi.org>

³³ <https://tools.ietf.org/html/rfc2119>

SHOULD NOT	neměl by
MAY	může, je možný

Tabulka č. 3 - České ekvivalenty klíčových slov z RFC2119

Pro definici struktury ukládaných dat metodika využívá doménového modelu, který je zvláštním případem diagramu tříd grafického jazyka UML specifikace verze 2.5.1³⁴.

6.4.2 Vkládání (tvorba) a aktualizace obsahu

1. Obsah **musí** být komponován z nezávislých bloků.
2. Jednotlivé bloky **by neměly** mít pevnou pozici. Uživatel **by měl** mít možnost jednoduše měnit jejich pořadí.
3. Náhled výsledného zobrazení **musí** být oddělen od editační části.
4. Obsah **musí** být tvořen z následujících základních bloků, které **musí** respektovat formát uložení uvedený v kapitole 6.4.3 Ukládání obsahu:
 - odstavec (třída Paragraph),
 - seznam (třída List),
 - tabulka (třída Table),
 - připojený obsah (třída Embedded content),
 - čisté HTML (třída HTML field; pro odborné uživatele).
5. Z výše definovaných bloků **by měla** implementující aplikace umět vytvářet skupiny pro složitější obsahové prvky (např. fotogalerie).
6. **Je možné** připojení externího obsahu, který je specifický pro implementující aplikaci (není součástí vytvářeného obsahu).
 - Náhled výsledného zobrazení **by měl** být zobrazen v reálném čase.
 - Aplikace implementující metodiku **by měla** umožňovat vytvářet šablony obsahu.
 - Šablony **mohou** nastavit pevnou pozici bloků (nebo skupin bloků) a tvořit tak výjimku z pravidla č. 2.

³⁴ <http://www.omg.org/spec/UML>

- Šablony **by neměly** být primárně statické a uživatelé **by měly** být stále schopni ve výsledném obsahu měnit pořadí bloků.
- Šablony **by měly** být přizpůsobeny pro automatický metadatový popis bez nutnosti zásahu autorů obsahu.

6.4.3 Ukládání obsahu

1. Implementující aplikace **musí** obsah ukládat tak, aby odpovídal Doménovému modelu Diagramu tříd UML, který ukazuje Diagram č. 4.

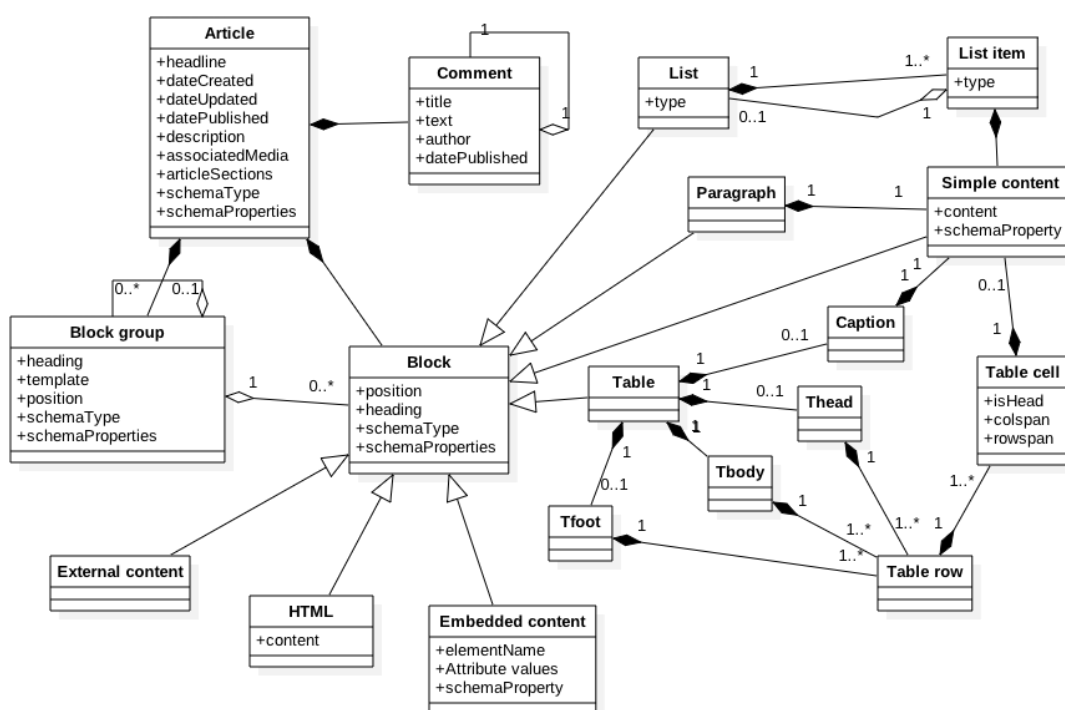


Diagram č. 4 - Model uložení obsahu; zdroj: vlastní zpracování

Popis tříd a atributů

Article

Základní třída nesoucí instanci informačního obsahu (např. článek). Označení Article vychází z definice elementu <article> specifikace HTML5.

Atribut	Popis
headline	Hlavní nadpis/název.
dateCreated	Datum, kdy byl obsah vytvořen (poprvé uložen).
dateUpdated	Datum poslední úpravy (posledního uložení).
datePublished	Datum, kdy byl obsah publikován. Zpravidla se zobrazuje čtenářům obsahu a je součástí metadatových výstupů.
description	Úvodní odstavec (perex), který se zobrazuje v různých výpisech, metadatových výstupech, ve výsledcích vyhledávání apod.
associatedMedia	Úvodní obrázek, jeho použití je nepovinné, závisí na implementační aplikaci.
articleSections	Zařazení článku do tematické kategorie, nebo kategorií.
schemaType	Určuje typ obsahu – schema (objekt) ze slovníku schema.org.
schemaProperties	Obsahuje asociativní pole nesoucí dodatečná metadata, která nelze automaticky odvodit z ostatních částí obsahu.

Tabulka č. 4 - Metodika WICM, atributy třídy Article; zdroj: vlastní zpracování

Block

Třída Block obsahuje je základní jednotkou (blok) strukturovaného obsahu. Nese atomizovaný obsah. Bloky lze pro složitější obsahové prvky seskupovat do skupin bloků (třída Block Group).

Atribut	Popis
position	Označuje pozici bloku ve skupině nebo samostatně. Podle pozice se bloky řadí pro vykreslení nebo editaci.
heading	Nepovinně může být blok doplněn nadpisem. Úroveň nadpisu (h1 – h6) se řídí automaticky v redakčním systému tak, aby byla sémanticky správně.
schemaType	Určuje typ obsahu bloku – schema (objekt) ze slovníku schema.org

schemaProperties	Obsahuje asociativní pole nesoucí dodatečná metadata, která nelze automaticky odvodit z ostatních částí obsahu.
-------------------------	---

Tabulka č. 5 - Metodika WICM, atributy třídy Block; zdroj: vlastní zpracování

Block Group

Třída Block Group slouží pro vytváření složitějších obsahových prvků (struktur) komponováním ze základních bloků (třídy Block), nebo skupin bloků (třídy Block Group). Slouží také pro hierarchické strukturování a tím určení úrovně nadpisů (h1 – h6) z atributů heading.

Atribut	Popis
position	Označuje pozici v rámci skupiny, do které patří, nebo třídy Article. Podle pozice se bloky řadí pro vykreslení nebo editaci.
heading	Skupina bloků by měla být (ale nemusí) doplněna nadpisem. Úroveň nadpisu (h1 – h6) se řídí automaticky v redakčním systému tak, aby byla sémanticky správně.
template	Pravidlo pro zobrazení. Jak se mají jednotlivé podskupiny nebo bloky složit dohromady. Definice šablony je uvedena v kapitole 6.4.4.
schemaType	Určuje typ obsahu bloku – schema (objekt) ze slovníku schema.org
schemaProperties	Obsahuje asociativní pole nesoucí dodatečná metadata, která nelze automaticky odvodit z ostatních částí obsahu.

Tabulka č. 6 - Metodika WICM, atributy třídy Block Group; zdroj: vlastní zpracování

List

Třída list je určena pro seznamy – viz kapitoly 4.4.5, 4.4.6, 4.4.7 specifikace HTML5.

Atribut	Popis	Přípustné hodnoty
type	Určuje, o jaký typ seznamu se jedná.	ol – třídění seznam ul – netříděný seznam dl – seznam definic

Tabulka č. 7 - Metodika WICM, atributy třídy List; zdroj: vlastní zpracování

List Item

Položka v seznamu. Položka může obsahovat jednak další seznam (zanoření), nebo textový obsah. Ten je dán třídou Simple Content.

Atribut	Popis	Přípustné hodnoty
type	Určuje, o jaký typ položky seznamu se jedná.	li – pro seznamy typu ol a ul dt – pro seznam typu dl dd – pro seznam typu dl

Tabulka č. 8 - Metodika WICM, atributy třídy List Item; zdroj: vlastní zpracování

Paragraphs

Třída pro uchovávání obecného textu ve formě odstavců, tedy elementů p – viz kapitola 4.4.1 specifikace HTML5.

Třída Paragraphs nemá žádné atributy.

Table, Caption, Thead, Tbody, Tfoot, Table Row

Třída Table slouží pro strukturované uchování obsahu tabulek – viz kapitola 4.9 specifikace HTML5. Tabulka nemusí obsahovat všechny navázané třídy (tedy sémantické elementy). Povinná je pouze třída Tbody a Table Row. Struktura se primárně řídí pravidly uvedenými ve specifikaci HTML5.

Třídy Table, Caption, Thead, Tbody, Tfoot, Table Row nemají žádné atributy.

Table Cell

Třída pro uchování jednotlivých buněk tabulky. Její obsah je definován ve třídě Simple Content.

Name	Description
isHead	Indikuje, zda se má pro buňku použít element th (hlavička tabulky).

colspan	Určuje rozšíření dané buňky do více buněk v horizontálním směru (pro spojování buněk).
rowspan	Určuje rozšíření dané buňky do více buněk ve vertikálním směru (pro spojování buněk).

Tabulka č. 9 - Metodika WICM, atributy třídy *Table Cell*; zdroj: vlastní zpracování

Simple content

Třída pro uchovávání nestrukturovaného textového obsahu. Může obsahovat holý text a případně elementy ze sekce 4.5 specifikace HTML5 (Text-level semantics).

Atribut	Popis
content	Samotný textový obsah.
schemaProperty	Název vlastnosti (property) ze slovníku schema.org, pokud tomu odpovídá význam obsahu atributu <i>content</i> této třídy.

Tabulka č. 10 - Metodika WICM, atributy třídy *Simple content*; zdroj: vlastní zpracování

Embedded content

Třída *Embedded content* slouží pro uchovávání např. obrázků, videí a podobného obsahu definovaného v sekci 4.7. specifikace HTML5.

Atribut	Popis
elementName	Název elementu identifikující typ obsahu.
Attribute values	Názvy a hodnoty atributů náležící k danému elementu.
schemaProperty	Název vlastnosti (property) ze slovníku schema.org, pokud tomu odpovídá význam obsahu této třídy.

Tabulka č. 11 - Metodika WICM, atributy třídy *Embedded content*; zdroj: vlastní zpracování

HTML

Třída pro ukládání obsahu ve formě HTML. V některých případech by uživatelé měli mít možnost vložit přímo obsah ve formátu HTML, případně doplněný o skripty (např. JavaScript).

Atribut	Popis
<code>content</code>	Samotný obsah ve formátu HTML.

Tabulka č. 12 - Metodika WICM, atributy třídy HTML; zdroj: vlastní zpracování

External content

Tento blok odkazuje na obsah, který je specifický pro implementační aplikaci. Může načítat data z jiné části aplikace, interaktivní prvky apod. V redakčních systémech se často odkazuje např. na podobné články, pokračování seriálů, fotogalerii, část textu z jiného článku, anketu, formulář apod. Tento obsah může být často interaktivní a jeho konkrétní podoba se může měnit. O finální prezentaci se tak stará aplikace, která obsah zobrazuje.

Třídá External content nemá žádné atributy.

Comment

Třída slouží pro uchovávání komentářů k obsahu.

Atribut	Popis
<code>title</code>	Samotný obsah ve formátu HTML.
<code>text</code>	Titulek komentáře, nepovinně.
<code>author</code>	Jméno autora, případně odkaz na jeho profil.
<code>datePublished</code>	Datum publikování komentáře.

Tabulka č. 13 - Metodika WICM, atributy třídy Comment; zdroj: vlastní zpracování

6.4.4 Prezentace, sémantika a archivace

Prezentace obsahu ve webovém prohlížeči

1. Každý prvek z třídy Block **musí** obalen elementem, který vychází z konkrétní třídy a specifikace HTML5. Blok *HTML field* a *External content* je uzavřen elementem `<div>`.

2. Každý prvek třídy Block group se **musí** skládat z prvků třídy Block nebo Block group. Sestavení je určeno šablonou HTML (atribut template).
3. Každá instance obsahu (třída Article) **musí** být sestavena z prvků tříd Block a Block group. Řazení vychází z atributu position. Obsah je obalen elementem <article>. Základní informace, které nese třída Article, **by měly** být obaleny v elementu <header>.
4. Obsah **musí** být sestaven minimálně z následujících elementů:
 - <header> (základní informace o obsahu)
 - i. <hX>
 1. Nadpis celého obsahu.
 2. Úroveň nadpisu vychází vždy ze struktury stránky, kde je článek vkládán.
 3. Vždy jen jednou, podnadpisy by dle (W3C, 2014a) neměli být v elementu h.
 - ii. <time>
 1. Datum publikování obsahu.
 2. Může být doplněno o další elementy time, např datum úpravy apod.
 - iii. <p>
 1. Úvodní odstavec (perex).
 - <section>
 - i. Tělo obsahu.
 - ii. Obsahuje vše ze tříd Block a Block group definovaných v kapitole 6.4.3.
 - <footer>
 - i. Informace o autorovi a další doplňující informace.
5. HTML výstup **by měl** být doplněn metadatovým popisem ve formátu RDFa, JSON-LD nebo Microdata s využitím některého ze slovníků iniciativy schema.org.
6. Doporučení metadatového slovníku se **může** v budoucnu změnit.

Specifikace šablony (template)

- Šablona **musí** obsahovat odkazy na jednotlivé bloky (nebo skupiny bloků) syntakticky oddělené složenými závorkami („{“, „}“).
- Odkaz **může** být ve formátu {block}. Při vykreslování jsou pak tyto odkazy nahrazovány postupně dle pořadí z atributů position.
- Šablona **může** obsahovat html elementy.

Archivace a přenos obsahu

1. Implementující aplikace **musí** poskytovat možnost exportu do formátu dle JSON Schematu <http://kit.pef.czu.cz/JSONschemas/wicm.json>.
2. Implementující aplikace **by měla** používat formát z bodu 1 pro přenos dat mezi serverovou a klientskou částí při prezentaci obsahu (pokud se nepoužívá již vygenerovaný kód v jazyce HTML).
3. Budoucí verze metodiky **musí** být vždy zpětně kompatibilní se staršími exportními formáty, aby bylo možné přistupovat i k archivovanému obsahu.

JSON schema z bodu 1 a všechna asociovaná schémata jsou uvedeny v příloze. Základní wicm.json schema uvádí Zdrojový kód č. 4.

```
{
  "$id": "http://kit.pef.czu.cz/JSONschemas/wicm.json",
  "$schema": "http://json-schema.org/draft-06/schema#",
  "type": "object",
  "properties": {
    "headline": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "dateCreated": {
      "type": "string",
      "format": "date-time"
    },
    "dateUpdated": {
      "type": "string",
      "format": "date-time"
    },
    "datePublished": {
```

```

    "type": "string",
    "format": "date-time"
  },
  "associatedMedia": {
    "type": "object",
    "$ref": "http://kit.pef.czu.cz/JSONschemas/embedded_schema.json"
  },
  "articleSections": {
    "type": "array",
    "items": {
      "type": "string"
    }
  },
  "blocks": {
    "type": "array",
    "items": {
      "type": "object",
      "$ref": "http://kit.pef.czu.cz/JSONschemas/block_schema.json"
    }
  },
  "blockGroups": {
    "type": "array",
    "items": {
      "type": "object",
      "$ref": "http://kit.pef.czu.cz/JSONschemas/blockGroup_schema.json"
    }
  },
  "schemaType": {
    "type": "string"
  },
  "schemaProperties": {
    "type": "array",
    "minItems": 0,
    "items": {
      "type": "object",
      "properties": {
        "propertyName": {
          "type": "string"
        },
        "propertyValue": {
          "type": "string"
        }
      }
    }
  },
  "required": [
    "headline",
    "dateCreated"
  ]
}

```

Zdrojový kód č. 4 Základní JSON Schema

7 Ověření, zhodnocení přínosů a možností metodiky

Metodika WICM byla průběžně ověřována při testování vyvíjeného prototypu. Uživatelská část metodiky byla ověřena pomocí metody Focus Groups. Další aspekty metodiky byly, vzhledem k její povaze, pro ověření konzultovány s odborníky z praxe formou řízených rozhovorů. Pro další ověření by bylo nutné metodiku implementovat a měřit např. výsledky webových stránek pomocí analytických nástrojů. To bude vzhledem k časové náročnosti na vývoj podrobené dalšímu zkoumání.

7.1 Ověření pomocí metody Focus Groups

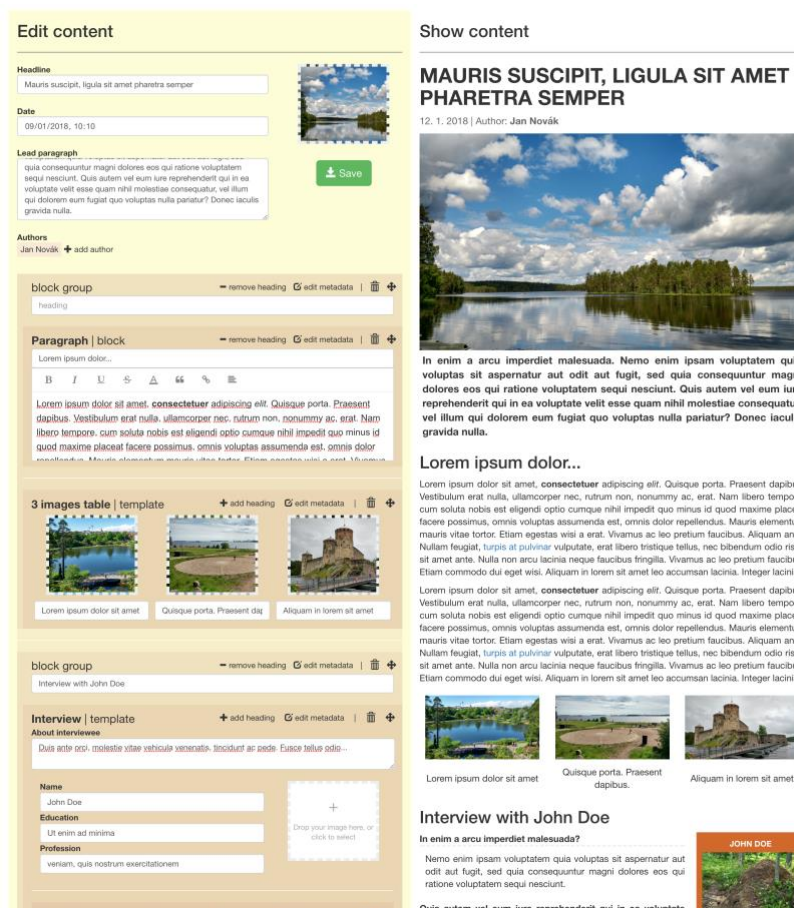
Prototyp, který byl vyvíjen během návrhu metodiky, byl představen stejné skupině uživatelů jako v počátečních fázích návrhu (viz. kapitola 6.3.3). Během sezení byla nejprve účastníkům prezentována navržená metodika, následně představen výsledný prototyp (viz. Obrázek č. 22 - Náhled vyvinutého prototypu), uživatelé si jej mohli vyzkoušet vložením několika modelových článků. Na závěr byl proveden rozhovor ohledně přínosů jeho používání, zhodnocení přínosů oproti současnému stavu a návrhu na další rozvoj metodiky.

V rámci aplikace metody Focus Groups byly sumarizovány nejdůležitější poznatky od uživatelů. Zjištěné poznatky a požadavky byly podrobeny kritické analýze a zhodnocena jejich důležitost a relevantnost. Signifikantní zjištěné poznatky a požadavky jsou následující:

- Uživatelé kladně hodnotili automatické ukládání. Nutnost neustálého ukládání je jeden z faktorů, proč stále tvoří raději obsah v textovém procesoru a až poté jej vkládají do redakčního systému.
- Někteří uživatelé by uvítali možnost pracovat bez připojení k internetu. Tato možnost nebyla v prototypu nijak řešena. Je však možné tuto funkcionalitu doporučit pro implementující aplikace a reakční systémy.
- Nutnost obsah strukturovat je v počátku pro uživatele obtížnější. Možnost vytvářet obsah s pomocí šablon na druhé straně vytváření obsahu značně zjednodušuje a pomáhá jim jej vytvořit kvalitnější.

- Během testování s uživateli byli identifikovány dílčí problémy s použitelností. Testován však byl prototyp, ne výsledná aplikace. Lze tak doporučit při implementaci provést testování s uživateli konkrétní cílové skupiny a přizpůsobit editaci obsahu jejich potřebám.
- Některým uživatelům nebyla příliš jasná funkcionalita přidávání metadatového popisu. Bylo diskutováno, že tento proces by ve výsledné aplikaci měl být více automatizovaný. Např. předpřipravenými schématy již v šabloně tak, aby tím uživatel nebyl rušen a mohl se soustředit víc na obsah viditelný čtenářům.

Uživatelé hodnotili celkově představený prototyp kladně. Konstatovali, že při dobré definici šablon by navržený koncept (metodika) pro ně byl přínosem.



Obrázek č. 22 - Náhled vyvinutého prototypu – část 1/2; zdroj: vlastní zpracování



Obrázek č. 23 - Náhled vyvinutého prototypu – část 2/2; zdroj: vlastní zpracování

7.2 Řízení rozhovory s odborníky s praxe

Pro další ověření navržené metodiky byla, vzhledem k její charakteristice časové a zdrojové náročnosti implementace, zvolena metoda polostrukturovaných rozhovorů s několika odborníky z praxe. Byli osloveni odborníci z České televize, Seznam.cz a několik nezávislých vývojářů a analytiků. Byl jim představen vyvinutý prototyp i navržená metodika a proveden rozhovor na základě následující osnovy.

Osnova polostrukturovaných rozhovorů

Základní otázky pro rozhovory byly následující:

- Jak hodnotíte možnost práce s obsahem pro běžné uživatele – strukturování obsahu?
- Je pro vás navržený model uložení dostatečný a zahrnuje všechny potřebné formy obsahu?
- Pomůže podle vás navržená standardizace rychlejšímu vývoji?
- Jak řešíte metadatový popis a byl pro vás návrh řešení v metodice přínosem?
- Jak řešíte aktualizace redakčních systémů? Může v tomto ohledu být exportní formát přínosem?
- Jak hodnotíte celkový přínos metodiky a jak vidíte možnosti širší adopce a implementace?

Shrnutí a zhodnocení odpovědí

Respondenti se většinou shodovali, že navržený postup práce s obsahem odpovídá současným trendům. Ve strukturování obsahu vidí přínos, nový směr a dobré řešení nedostatků WYSIWYG editorů. Současně však upozorňovali na to, že prototyp má pravděpodobně několik nedostatků z hlediska použitelnosti a bylo by dobré při implementaci provést důkladné testování s uživateli s pokročilejších metod. Upozorňovali například na nedostatky implementace metadatového popisu. Uživatelé by k tomu měli být vedeni více názorně a popis by měl být v maximální míře automatický skrze šablony.

Model uložení považují respondenti za dostatečný, avšak současně upozorňovali na fakt, že by měla být finální metodika konzultována s open source komunitou, zejména pak s výrobcí redakčních systémů. Rozšíření navrženého modelu by spatřovali především v navržených attributech. Bylo by patrně možné najít další možné atributy pro jednotlivé třídy. Zároveň by bylo vhodné definovat nutnost jejich použití – povinné, nepovinné. Je však otázka, nakolik jsou jednotlivé atributy závislé na konkrétní implementaci.

Největší přínos navržené metodiky pak vidí respondenti ve standardizaci uložení. Respondenti poukazovali zejména na fakt, že open source a standardizace značně zrychlují vývoj software. Dnes se vývojáři téměř neobejdou bez knihoven a frameworků, které řeší zaběhnuté a opakující se funkcionality. Opět pak upozorňovali na nutnost diskuze v rámci open source komunity pro širší adopci standardu. Standardy jsou v prostředí WWW většinou přijímány na základě konsensu. Dále respondenti poukazovali na možnost rozšíření standardu směrem k implementaci používanými databázovými systémy – např. relační MySQL nebo NoSQL MongoDB.

Při rozhovorech bylo možné zaznamenat shodu nad aktuálností oblasti sémantiky a metadatového popisu. Navržené rozšíření možností stávajících nástrojů vidí jako přínosné. Největší problém pak vidí v tom, že je často pro metadatový popis potřeba aktivita uživatelů (autorů obsahu) pracujících s obsahem. Je tak nutné je k tomu vést a není možné nechat metadatový popis čistě na nich. Po širší diskuzi nad tématem metadatového popisu někteří respondenti souhlasili s nedostatky metadatových slovníků směrem k hlubšímu popisu informačního obsahu.

U navrženého exportního formátu se respondenti shodovali na tom, že jeho podoba je závislá především na modelu uložení. Nejvíce přínosný jej vidí v oblasti přenosu dat pro mobilní zařízení, kde by umožnil vývoj knihoven pro aplikace mobilních zařízení. Z hlediska problémů vznikajících během aktualizací redakčních systémů pak nevidí respondenti takový problém. Pro pokročilejší webové portály se nejčastěji používá redakční systém Drupal (např. Česká televize), kde se přechod na novější verzi realizuje v řádu let. Redakční systém WordPress je pak vyvíjen kontinuálně a změny jsou uváděny také kontinuálně. Přínosnější pak může být navržený formát pro přechod na pokročilejší redakční systém.

Celkově odborníci hodnotili metodiku jako přínosnou. Největší riziko pak spatřují v implementaci do stávajících redakčních systémů. Navíc je pro implementaci nutná shoda napříč open source komunitou. Vzhledem k vývoji v oblasti webových technologií je však možné pozorovat nově vznikající redakční systémy, které využívají

moderní přístupy k tvorbě webových aplikací (izomorfní přístup, NoSQL databáze). Zde je možnost adopce metodiky pravděpodobně vyšší.

7.3 Zhodnocení možných přínosů metodiky WICM

7.3.1 Přínosy pro autory obsahu

Uživatelům, kteří spravují informační obsah a nemají znalosti technologií jako je HTML, CSS nebo JavaScript, by metodika měla umožnit snadněji vkládat kvalitní obsah. Jednodušší by pro ně mělo být zejména vkládání složitějších částí obsahu, jež je v klasických WYSIWYG editorech komplikované. Díky omezením stanoveným novým Content modelem (viz. kapitola 6.3.3) by mělo docházet k menší chybovosti na vstupu. Tím by měly být omezeny problémy s konzistencí HTML, které vznikají u WYSIWYG editorů (viz. kapitola 5.1). Strukturovaný obsah může být zároveň responzivní, aniž by jej autoři byli nuceni k tomu explicitně upravovat. Uživatelé také mohou být metodikou vedeni ke správnému metadatovému popisu. Při kvalitní definici šablon pak může obsah sémantický i bez zásahu autorů. Možnost vytvářet obsah na základě předdefinovaných šablon pak pomáhá držet jednotný vzhled určitých typů obsahu. To pomáhá v závěru zejména pravidelným čtenářům, kteří se poté v obsahu snáze orientují.

7.3.2 Přínosy pro vývojáře

Standardizace a sjednocení, které předkládaná metodika přináší, může přinést zrychlení vývoje redakčních systémů a obecně také webových aplikací (viz. kapitola 6.2.2). Bude možné vyvíjet znovupoužitelné knihovny a frameworky a na nich stavět aplikace a redakční systémy, které metodiku implementují. Standardizace struktury pro ukládání a export může usnadnit aktualizace redakčních systémů, ale i přechod k jiným. To je vzhledem k aktuálnímu trendu v prostředí WWW, kdy se stále více využívá JavaScript na straně serveru a objevují se nové redakční systémy (viz. kapitoly 4.4.5 a 5.2), stále více aktuální. Z těchto výhod mohou těžit také vývojáři aplikací pro mobilní telefony. Standardizace opět umožní vývoj knihoven. Strukturování obsahu

pak může pomoci při práci s obsahem v zařízení, kde s ním mobilní aplikace mohou pracovat efektivněji.

7.3.3 Přínosy pro čtenáře obsahu

Pro uživatele – čtenáře obsahu je přínosem zejména kvalitnější obsah. Díky menší chybovosti na vstupu, konzistenci HTML a lepší podpoře responzivity se obsah bude zobrazovat lépe a bude pro ně lépe čitelný a použitelný. Obsah také může být díky strukturovanosti být dynamičtější a čtenáře tak více zaujmout. Čtenáři by například mohli být více zapojeni díky širším možnostem interakce, propojení se sociálními sítěmi apod.

7.3.4 Přínosy v oblasti sémantiky

Sémantika v současných nástrojích a redakčních systémech není dostatečně řešena (viz. kapitola 5). Navržená metodika umožňuje snazší metadatový popis informačního obsahu a tím i zlepšení jeho dostupnosti. Sémantika v prostředí WWW pomáhá primárně indexovacím nástrojům vyhledávačů. V poslední době stále více rozvíjí různé chytré asistenty a umělá inteligence obecně (viz. kapitola 4.5). Díky sémantice tak mohou obsahu lépe rozumět a dále s ním pracovat. V neposlední řadě lepší sémantika pomáhá také v oblasti přístupnosti.

7.3.5 Ekonomické přínosy

Pro firmy jako jsou např. mediální společnosti může mít metodika i značné ekonomické přínosy. Již samotné využití open source software může značně snížit náklady na vývoj vlastních softwarových řešení. Díky existenci otevřených knihoven mohou ušetřit osobní náklady na vývojáře. Standardizovaný formát pro přenos obsahu a jeho strukturovanost pak může pomoci při další práci s obsahem. Může pomoci např. při aktualizacích a přechodu na jiný redakční systém, nebo archivaci obsahu. Obsah může být např. snadněji transformován pro tisk apod.

Vzhledem k aktuální situaci na trhu práce, kde je nedostatek pracovníků v IT sektoru a tím i stále se zvyšující finanční ohodnocení, může být úspora mzdových nákladů velmi

významná. Informační obsah pak mohou vkládat i autoři obsahu jako jsou např. redaktoři. Podniky tak mohou ušetřit na odbornících, kteří se starají pouze o správu obsahu. Díky lepší kvalitě obsahu pak mohou zaujmout více čtenářů a tím zvýšit své příjmy.

7.4 Možnosti a doporučení pro další výzkum

Jednou z možností následného výzkumu je oblast sémantiky. Zde je možné spatřit výzkumný potenciál v oblasti metadatového popisu samotného informačního obsahu zejména z hlediska jeho ontologie. Nabízí se možnosti rozšíření metadatových slovníků vzhledem k rozšiřující se oblasti strojového učení, umělé inteligence a chytrých asistentů.

Další možností je rozšíření standardizace modelu uložení. Jak poukazovali odborníci z praxe během rozhovorů – bylo by vhodné stanovit povinnost použití atributů. Dále by bylo vhodné vytvořit návazné modely pro implementaci v nejvyužívanějších databázových systémech – relační MySQL nebo NoSQL MongoDB. To by mohlo pomoci pro vývoj knihoven a rozšíření pro redakční systémy.

Navrženou metodiku by v dalším výzkumu bylo možné upravit pro využití v řadě specializovaných oblastí. Nabízí se například možnost využití redakcemi vědeckých časopisů, které používají většinou šablony pro MS Word. Práce s nimi je pak často problematická a vyžaduje dodatečnou komunikaci mezi editory a autory ohledně formálních požadavků časopisu. Strukturovaný obsah by tak mohl validovat tyto formální požadavky již na vstupu. Zároveň by takový obsah mohl pomoci v automatizaci přípravy sazby pro tisk nejen pro redakce vědeckých časopisů, ale obecně všem vydavatelstvím.

Další výzkum by mohl pokračovat v ověření metodiky při její implementaci, kde by bylo možné kvantifikovat přínosy pomocí měření různých atributů v oblasti optimalizace pro vyhledávače, sémantiky, dostupnosti obsahu apod.

Pro širší rozšíření metodiky je potřeba se zaměřit na její popularizaci, která je klíčová pro budoucí implementace. Bylo by vhodné vyvinout knihovny

pro nejvyžívanější programovací jazyky a frameworky. Pro širší přijetí standardu bude potřeba další vývoj a konsensus v rámci open source komunit.

8 Závěry dizertační práce

Dílčí cíl č.1 - Prozkoumat současný stav poznání v oblasti „data, informace, znalosti“ a vymežit pojem „informační obsah“ v prostředí WWW

V kapitolách 4.1, 4.2 a 4.3 byly zkoumány a porovnávány různé definice pojmů data, informace, znalosti a prostředí World Wide Web (WWW) a to zejména v oblasti zaměření práce. Syntézou poznatků byl vymezen pojem informační obsah. Je to zpráva, která je předávána prostřednictvím internetového prohlížeče uživateli. Tato zpráva je sestavena z dat uložených na straně serveru, která již byla setříděna a kategorizována a k uživateli se dostávají v určité dané formě. Z hlediska sémantiky značkovacího jazyka HTML5 je to obsah, který může být uvnitř elementu `<article>`.

Dílčí cíl č.2 - Prozkoumat současný stav a rozvoj webových technologií vzhledem k zaměření hlavního cíle

V kapitole 4.4 byly podrobeny zkoumání současné webové technologie a vybrané způsoby jejich užití. Webové technologie se obecně dělí na serverové a klientské, kde se klientem se rozumí webový prohlížeč. Jednotlivé technologie se dnes nepoužívají samostatně a navzájem se doplňují. Dominantní technologií pro zobrazení webových stránek je značkovací jazyk HTML. Ten je doplněn kaskádovými styly (CSS), které upravují vzhled pro uživatele a také způsob a chování zobrazení webového obsahu na různých zařízeních. V současné době je v průměru přibližně polovina přístupů na internet realizována z mobilních zařízení a responzivita je tak jednou z klíčových vlastností webových stránek.

Na straně serveru je dominantní technologií stále PHP, které využívá většina webových stránek a aplikací. V prostředí WWW je však vidět posun od tradičního přístupu, kde je část logiky vykonávána jen na serveru a část ve webovém prohlížeči. Stále více se zde rozšiřuje prostředí Node.js a s ním i jazyk JavaScript na straně serveru. V posledních letech jsou stále více populární tzv. single-page aplikace, které přenášejí většinu logiky na stranu klienta. S rozvojem tohoto trendu i mezi klasické webové

stránky se dnes setkáváme s pojmem izomorfní aplikace. Ty využívají možnosti běhu stejného kódu na straně klienta i serveru.

Webové aplikace založené na jazyce PHP většinou využívají relační databázové systémy, nejčastěji MySQL. Stále více se však v prostředí WWW rozšiřují i databáze typu NoSQL, a to ve spojitosti s rozvojem serverového JavaScriptu a Node.js.

Dílčí cíl č.3 - Prozkoumat současný stav a rozvoj technologií a nástrojů pro sémantiku v prostředí WWW

Kapitola 4.5 se zabývala problematikou sémantiky v prostředí WWW. Webovému obsahu dnes kromě lidí potřebují rozumět i počítače a jejich programy. Historicky jsou to především indexovací nástroje vyhledávačů. V souvislosti s velmi rychlým rozvojem různých chytrých asistentů a umělé inteligence tato potřeba nabývá stále více na významu. Základní sémantika je obsažena již v samotném pojmenování elementů HTML5. To však není dostatečné. Webové stránky je tak potřeba ještě doplnit metadatovým popisem.

Specifikace HTML5 s tím počítá a zahrnuje přímou podporu pro Microdata. Microdata spolu s RDFa a JSON-LD jsou nejvyužívanější formáty pro metadatový popis. Vždy se používají ve spojení s metadatovým slovníkem, který popisuje použité objekty. V prostředí WWW se nejčastěji využívají slovníky iniciativy Schema.org.

Dalším využívaným formátem zápisu metadat je Open Graph. V rámci HTML využívá elementů *meta* v hlavičce dokumentu. Používá se pro popis webových objektů pro použití především v rámci sociální sítě Facebook.

Dílčí cíl č.4 - Analyzovat používané metody a nástroje pro sběr a zpracování dat v prostředí www vzhledem k zaměření hlavního cíle

Pro splnění tohoto dílčího cíle byly v kapitole 5 analyzovány současné metody a nástroje pro správu informačního obsahu. Kapitola vychází z obecného přehledu současného stavu této problematiky v prostředí WWW. Bylo zjištěno, že současné WYSIWYG editory jsou již funkčně nedostatečné a pomalu ztrácejí význam. Informační obsah není poskytován pouze prostřednictvím webových prohlížečů desktopových

počítačů, ale stále více je prohlížen i na mobilních zařízeních, kde musí být responzivní a přizpůsobovat se různým velikostem obrazovek a zařízení.

Dále byl v kapitole 5.2 zkoumán pojem Web Content Management System jako obecný název pro webové aplikace používané pro chod a správu webových stránek. Z tohoto pojmu pak vychází pojem Content Management System, česky Redakční systém, který je dnes používán širokou veřejností.

Většina současných redakčních systémů je stále technologicky založena na jazyce PHP a databázovém serveru MySQL. S rychlým rozvojem JavaScriptu a NoSQL databází se ale pomalu objevují i redakční systémy založené těchto technologiích. V budoucnu je tak možné očekávat růst jejich významu.

V kapitole 5.3 bylo zkoumáno, jakým způsobem přistupují současné nejvýznamnější redakční systémy (Wordpress, Joomla, Drupal) ke správě informačního obsahu a jak řeší vkládání, ukládání i prezentaci. Všechny redakční systémy přistupují v základu ke správě informačního obsahu rozdílně. Pomocí volitelných rozšíření je možné např. strukturovat obsah a skládat jej z nezávislých bloků. Nejdále je v tomto ohledu redakční systém Drupal při použití doplňku Paragraphs. Drupal má také jako jediný podporu strukturovaného obsahu již v jádře. Další doplňky jsou již jen nadstavby a nemění příliš výchozí správu obsahu. Situace u redakčního systému Joomla je v tomto případě na nejnižší úrovni. Zmiňované chování nabízí více doplňků a celkově je zde správa obsahu příliš fragmentovaná a chybí zde jednotný přístup. WordPress pak poskytuje robustní řešení, které je jednoduché pro uživatele, ale nenabízí tolik možností pro úpravy, jako výše zmíněné redakční systémy.

Žádný z výše uvedených redakčních systémů uspokojivě neřeší problematiku sémantiky a metadatového popisu. Není možné změnit šablonu výsledného HTML kódu obsahu a ovlivnit tak základní sémantiku, kterou nabízí HTML5. Nástroje pro práci s metadatovým popisem jsou implementována pouze na úrovni celé stránky a jsou určeny spíše pro pokročilé uživatele. Redakční systémy tak nenabízí nástroje (ani v rámci rozšíření) pro metadatový popis na úrovni informačního obsahu.

Na závěr kapitoly 5 jsou identifikovány výzkumné příležitosti pro návrh metodiky WICM ve třech oblastech: Vkládání a aktualizace obsahu (1); Ukládání obsahu (2); Prezentace, sémantika a archivace (3).

Hlavní cíl - Navrhnout metodiku pro vkládání, aktualizaci, uložení a prezentaci obsahu v prostředí World Wide Web

V kapitole 6 byla nejprve na základě předchozího textu definována hypotéza dizertační práce. Dále byly vymezeny cíle a požadavky na navrhovanou metodiku a byl vyvinut výchozí prototyp.

Metodika WICM byla formulována pomocí stejných principů, jako současné webové standardy a specifikace. Používá klíčová slova z doporučení IETF RFC2119, respektive jejich české ekvivalenty. Skládá se ze tří hlavních částí podle toho, kterou oblast řeší.

První část se zabývá oblastí **vkládání a aktualizace** obsahu. Obsahuje především doporučení, jaké by měli aplikace implementující metodiku používat nástroje a metody, jak strukturovat obsah a základní doporučení pro použitelnost.

V oblasti **ukládání** metodika stanovuje především obecnou strukturu ve formě doménového modelu diagramu tříd jazyka UML. Nejprve byl stanoven nový content model pro HTML, který omezuje strukturu na nejnižší úrovni a umožňuje tak jednodušeji přistupovat k ukládání obsahu. Následuje popis jednotlivých tříd doménového modelu a s tím spojených pravidel.

Třetí část se zabývá problematikou **prezentace, sémantiky a archivace**. Byly stanoveny základní pravidla pro zobrazení obsahu ve webovém prohlížeči. Byla definována základní syntaxe šablony pro zobrazování skupin bloků obsahu. Dále byl pomocí nástroje JSON Schema navržen exportní formát, který je určen pro přenositelnost obsahu, sdílení, archivaci ale i zpracování v dalších aplikacích a systémech. To mohou být např. aplikace pro mobilní zařízení nebo systémy zpracovávající obsah pro tisk.

Dílčí cíl č. 5 - Po splnění hlavního cíle ověřit navrženou metodiku

Část metodiky, určená pro uživatele, byla ověřena pomocí metody Focus Groups (viz. kapitola 7.1). V rámci sezení byla představena finální verze metodiky a byl demonstrován prototyp, který vznikl v rámci jejího návrhu. Uživatelé si vyzkoušeli použití prototypu na několika modelových článcích. Z následné diskuze byly identifikovány významné poznatky a formulována některá dílčí doporučení.

Dále bylo pro ověření metodiky provedeno několik polostrukturovaných rozhovorů s odborníky z praxe (viz kapitola 0). Respondentům byla představena metodika a byly diskutovány základní poznatky z vývoje a její nejdůležitější aspekty. Na základě syntézy výsledků z těchto rozhovorů byla metodika shledána jako přínosná. Dále byly formulovány doporučení pro budoucí výzkum a rozvoj metodiky. Byly tak ověřeny definované cíle a požadavky metodiky. Vzhledem k charakteristice metodiky však její další ověření vyžaduje dlouhodobější nasazení v praxi – např. v rámci redakčních systémů.

Dílčí cíl č. 6 - Zhodnotit možné přínosy navržené metodiky

V kapitole 7.3 byly diskutovány možné přínosy metodiky WICM pro různé cílové skupiny. Pro autory obsahu je to zejména zlepšení a zjednodušení práce. Za jeden z nejvýznamnějších přínosů je možné považovat standardizaci v oblasti informačního obsahu v prostředí WWW, která zde zcela chybí. To napomůže vývojářům k vytvoření veřejně dostupných knihoven a frameworků, které mohou v konečném důsledku zrychlit vývoj redakčních systémů a dalších webových nebo mobilních aplikací. Čtenářům obsahu pak metodika může přinést kvalitnější a dostupnější obsah, který je zároveň plně responzivní. Dále metodika řeší problematiku sémantiky, která není v současných redakčních systémech uspokojivě řešena. Sémantika pomáhá počítačům a jejich programům rozumět informačnímu obsahu. V neposlední řadě může mít metodika ekonomické přínosy ve formě snížení nákladů na vývoj i zvýšení příjmů díky kvalitnějšímu obsahu.

Dílčí cíl č. 7 - Synteticky zhodnotit dosažené výsledky a formulovat závěry a doporučení pro další výzkum

V práci byl nejprve vymezen pojem informační obsah na základě analýzy stavu poznání v oblasti data, informace, znalosti a prostředí World Wide Web. Dále byly analyzovány současné webové technologie, sémantika a redakční systémy jako nástroje pro sběr a zpracování dat v prostředí WWW. Byly analyzovány možnosti, které redakční systémy a jejich rozšíření nabízejí pro správu informačního obsahu. Syntézou poznatků z těchto analýz byla stanovena hypotéza dizertační práce a v návaznosti také cíle a požadavky na navrženou metodiku WICM. Metodika byla navržena obdobně jako současné webové standardy a specifikace. Dále byla metodika ověřena, byly zhodnoceny její přínosy a možnosti pro další výzkum. Navazující výzkum se může např. zaměřit na problematiku metadatového popisu na úrovni informačního obsahu, nebo např. rozšíření metodiky pro implementace v konkrétních databázových systémech.

9 Seznam použitých zdrojů

ARNOWITZ, Jonathan, Michael ARENT a Nevin BERGER, 2006. *Effective Prototyping for Software Makers* [online]. Burlington, UNITED STATES: Elsevier Science. ISBN 9780080468969. Dostupné

z: <http://ebookcentral.proquest.com/lib/czup/detail.action?docID=285771>

ARSENAULT, Cody, 2017. *OOCSS - The Future of Writing CSS* [online] [vid. 2018-01-13]. Dostupné z: <https://www.keycdn.com/blog/oocss/>

BALLY, Laurent, John BRITTAN a Karl H. WAGNER, 1977. A prototype approach to information system design and development. *Information & Management* [online]. **1**(1), 21–26. ISSN 03787206. Dostupné z: doi:10.1016/0378-7206(77)90005-2

BARCLAY, Kenneth a John SAVAGE, 2003. *Object-Oriented Design with UML and Java* [online]. Oxford, UNKNOWN: Elsevier Science. ISBN 9780080497556. Dostupné z: <http://ebookcentral.proquest.com/lib/czup/detail.action?docID=297134>

BERNERS-LEE, Tim, 1997. *Axioms of Web Architecture: Metadata* [online]. Dostupné z: <https://www.w3.org/DesignIssues/Metadata.html>

BERNERS-LEE, Tim a D CONNOLLY, 1995. *Hypertext Markup Language - 2.0* [online]. 1995. September. Dostupné z: http://www.w3.org/MarkUp/html-spec/html-spec_toc.html

BERNERS-LEE, Tim, James HENDLER a Ora LASSILA, 2001. The Semantic Web. *Scientific American* [online]. Dostupné z: <https://pdfs.semanticscholar.org/566c/1c6bd366b4c9e07fc37eb372771690d5ba31.pdf>

BERTINO, Elisa, Elena FERRARI, Andrea PEREGO a Gian Piero ZARRI, 2008. Advanced Techniques for Web Content Filtering. In: *Encyclopedia of Internet Technologies and Applications* [online]. B.m.: IGI Global, s. 36–44. Dostupné z: doi:10.4018/978-1-59140-993-9.ch006

BOIKO, Bob, 2005. *Content management bible*. 2nd ed. Indianapolis, IN: Wiley Pub. ISBN 07-645-7371-3.

BRÁZA, Jiří, 2005. *PHP 5*. 1. vyd. Praha: Grada. ISBN 80-247-1146-X.

BREHM, Spike, 2013. Isomorphic JavaScript: The Future of Web Apps – Airbnb Engineering & Data Science – Medium. *Medium.com* [online]. Dostupné z: <https://medium.com/airbnb-engineering/isomorphic-javascript-the-future-of-web-apps-10882b7a2ebc>

BROWN, Mandy, 2014. Writing and Editing in the Browser. *The Journal of Electronic Publishing* [online]. **17**(1). ISSN 1080-2711. Dostupné z: doi:10.3998/3336451.0017.111

CADLE, James, Tahir AHMED, Julian COX, Lynda GIRVAN, Alan PAUL a Debra PAUL, 2014. *Developing Information Systems : Practical guidance for IT professionals* [online]. Swindon, UNKNOWN: BCS Learning & Development Limited. ISBN 9781780172460. Dostupné

z: <http://ebookcentral.proquest.com/lib/czup/detail.action?docID=1713962>

CANAZZA, Mário Rodrigo, 2009. GLOBAL EFFORT ON BRIDGING THE DIGITAL DIVIDE AND THE ROLE OF ICT STANDARDIZATION. In: *ITU-T Kaleidoscope: Innovations for Digital Inclusions - K-IDI 2009*. ISBN 9261128912.

ČÁPKA, David, 2013. *UML - Doménový model* [online]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-domenovy-model-diagram>

CEJPEK, Jiří, 2005. *Informace, komunikace a myšlení*. 2., přeprac. vyd. Praha: Karolinum. ISBN 80-246-1037-X.

CHYTKOVÁ, Dagmar a Michal ČERNÝ, 2012. Znalostní a informační management. *Inflow Information Journal* [online]. Dostupné z: <http://www.inflow.cz/znalostni-informacni-management>

CROCKFORD, Douglas, 2006. *The application/json Media Type for JavaScript Object Notation (JSON)* [online]. 2006. Dostupné z: <http://www.ietf.org/rfc/rfc4627>

DAS, Sudeshna, Mark GOETZ, Lisa GIRARD a Tim CLARK, 2009. Scientific publications on web 3.0. In: *ELPUB 2009 - Rethinking Electronic Publishing: Innovation in Communication Paradigms and Technologies - Proceedings of the 13th International Conference on Electronic Publishing* [online]. s. 107–129. Dostupné z: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84864867845&partnerID=40&md5=aaf22a5286a9eca030c3f6956c29f25b>

DAVENPORT, Laurence a Thomas PRUSAK, 2000. *Working knowledge*. Boston, Mass.: Harvard business school press. ISBN 978-1-57851-301-7.

DESCARTES, René, 1992. *Rozprava o metodě*. 3.vyd., 1. Praha: Svoboda. ISBN 80-205-0216-5.

ECKSTEIN, Robert a Michel CASABIANCA, 2001. *XML pocket reference*. 2nd ed. Sebastopol, CA: O'Reilly. ISBN 05-960-0133-9.

ECMA, 2015. *ECMAScript 2015 Language Specification* [online]. 2015. 6th Editio. Ženeva, Švýcarsko: Ecma International. Dostupné z: <http://www.ecma-international.org/ecma-262/6.0/index.html>

FACEBOOK, 2017. *The Open Graph protocol* [online]. Dostupné z: <http://ogp.me/>

FAWCETT, Joe, Liam R QUIN a Danny AYERS, 2012. *Beginning XML*. 5th ed. Indianapolis, Ind.: John Wiley. ISBN 978-1-118-16213-2.

FORD, Jerry Lee, 2009. *Ajax programming for the absolute beginner*. Boston, MA: Course Technology Cengage Learning. ISBN 978-1-59863-564-5.

GARRETT, Jesse James, 2005. *Ajax: A New Approach to Web Applications* [online]. 2005. Dostupné z: <https://web.archive.org/web/20080702075113/http://www.adaptivepath.com/deas/essays/archives/000385.php>

GASSTON, Peter, 2014. *The Book of CSS3, 2nd Edition: A Developer's Guide to the Future of Web Design* [online]. B.m.: No Starch Press. ISBN 1593275803. Dostupné z: <https://www.amazon.com/Book-CSS3-2nd-Developers-Future/dp/1593275803?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1593275803>

GILMORE, W, 2005. *Velká kniha PHP 5 a MySQL*. Vyd. 1. Brno: Zoner Press. ISBN 80-868-1520-X.

GOLDSTEIN, Alexis, Louis LAZARIS a Estelle WEYL, 2011. *HTML5 & CSS3 for the Real World*. 1st ed. Collingwood, VIC, Australia: SitePoint. ISBN 978-0-9808469-0-4.

GOODMAN, Danny, 2010. *JavaScript Bible*. 7th ed. Indianapolis, Ind.: Wiley Pub., Inc. ISBN 978-0-470-52691-0.

GOOGLE, 2015. *Make sure your site's ready for mobile-friendly Google search results - AdSense Help* [online] [vid. 2018-01-13]. Dostupné z: <https://support.google.com/adsense/answer/6196932?hl=en>

HAROLD, Elliotte Rusty a W Scott MEANS, 2004. *XML in a Nutshell, Third Edition* [online]. B.m.: O'Reilly Media. ISBN 0596007647. Dostupné z: <https://www.amazon.com/Nutshell-Third-Elliotte-Rusty-Harold/dp/0596007647?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0596007647>

HENDL, Jan, 2008. *Kvalitativní výzkum*. 2., aktual. Praha: Portál. ISBN 978-80-7367-485-4.

HENDLER, Jim a Tim BERNERS-LEE, 2010. *From the Semantic Web to social machines: A research challenge for AI on the World Wide Web* [online]. 2010. ISBN 0004-3702. Dostupné z: doi:10.1016/j.artint.2009.11.010

HLAVENKA, Jiří, 1997. *Výkladový slovník výpočetní techniky a komunikací*. 3. vyd. Praha: Computer Press. ISBN 80-722-6023-5.

HOCK-CHUAN, Chua, 2009. *In Introduction to HTTP Basics* [online]. Dostupné z: https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html

JAVED, Ashar a Jörg SCHWENK, 2015. Systematically Breaking Online WYSIWYG Editors. In: [online]. s. 122–133. Dostupné z: doi:10.1007/978-3-319-15087-1_10

JENKINS, Kris, 2015. *What Is Functional Programming?* [online]. Dostupné z: <http://blog.jenkster.com/2015/12/what-is-functional-programming.html>

JSON.ORG, 2018. *Introducing JSON* [online]. 2018. Dostupné z: <http://json.org/json-cz.html>

K. ZYP, Ed. a G. COURT, 2013. JSON Schema: core definitions and terminology. *Internet Engineering Task Force* [online]. 14. Dostupné z: <http://tools.ietf.org/pdf/draft-zyp-json-schema-04.pdf>

KATUŠČÁK, Dušan, Marta MATTHAEIDESOVÁ a Marta NOVÁKOVÁ, 1998. *Informačná výchova*. 1. vyd. Bratislava: Slovenské pedagogické nakladateľstvo. ISBN 80-080-2818-1.

KEITH, Jeremy, 2010. *HTML5 for web designers*. New York: A book apart. ISBN 978-098-4442-508.

KHALILI, Ali a Søren AUER, 2015. WYSIWYM - Integrated visualization, exploration and authoring of semantically enriched un-structured content. *Semantic Web* [online]. 6(3), 259–275. ISSN 22104968. Dostupné z: doi:10.3233/SW-140157

KHALILI, Ali, Soren AUER a Daniel HLADKY, 2012. The RDFa content editor - From WYSIWYG to WYSIWYM. In: *Proceedings - International Computer Software and Applications Conference* [online]. s. 531–540. ISBN 9780769547367. Dostupné z: doi:10.1109/COMPSAC.2012.72

KOMENDA, Joe, 2017. The WYSIWYG is a Lie. *ThinkShout* [online]. Dostupné z: <https://thinkshout.com/blog/2017/08/the-wysiwyg-is-a-lie/%0D>

LAZARIS, Louis, 2011. An Introduction To Object Oriented CSS (OOCSS) — Smashing Magazine. *Smashing Magazine* [online] [vid. 2018-01-13]. Dostupné z: <https://www.smashingmagazine.com/2011/12/an-introduction-to-object-oriented-css-oocss/>

MALÝ, Martin, 2010. *Node.js – s JavaScriptem na server* [online]. 2010. Dostupné z: <http://www.zdrojak.cz/clanky/node-js-s-javascriptem-na-server/>

MALÝ, Martin, 2011. Single Page Apps a řešení problémů s historií - Zdroják. *Zdroják.cz* [online]. Dostupné z: <https://www.zdrojak.cz/clanky/single-page-apps-a-reseni-problemu-s-historii/>

MASNER, Jan, J. VANĚK a M. STOČES, 2014. Spatial data monitoring and mobile applications - Comparison of methods for parsing JSON in android operating system. *Agris On-line Papers in Economics and Informatics*. 6(1), 37–46. ISSN 18041930.

MASNER, Jan, Jiri VANEK a Jan JAROLIMEK, 2016. Prototype of a Content Creation and Updating Application Module for Agrarian Sector and Regional Development. In: *AGRARIAN PERSPECTIVES XXV: GLOBAL AND EUROPEAN CHALLENGES FOR FOOD PRODUCTION, AGRIBUSINESS AND THE RURAL ECONOMY*. DEPT SYSTEMS ENG, KAMYCKA 129, PRAGUE 6 165 21, CZECH REPUBLIC: CZECH UNIVERSITY LIFE SCIENCES PRAGUE, s. 206–213. Agrarian Perspectives Series. ISBN 978-8-0213-2670-5.

MASNER, Jan, Jiří VANĚK, Jan JAROLÍMEK a Vladimír OČENÁŠEK, 2015. Markup Languages Support for Content Management of Agricultural Portals. In: *HAICTA 2015*. Řecko: CEUR-WS, s. 594–602. ISSN 1613-0073.

MENING, Robert, 2015. WordPress vs. Joomla vs. Drupal (CMS Comparison). *WebsiteSetup.org* [online]. Dostupné z: <https://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>

MENING, Robert, 2017. Popular CMS & Market Share (2018) - WebsiteSetup.org. *WebsiteSetup* [online]. Dostupné z: <https://websitesetup.org/popular-cms/>

MERILINNA, J. a M. MATINLASSI, 2006. State of the Art and Practice of OpenSource Component Integration. In: *32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO'06)* [online]. s. 170–177. ISBN 0-7695-2594-6. Dostupné z: doi:10.1109/EUROMICRO.2006.61

MICROFORMATS2, 2017. *microformats2* [online] [vid. 2017-01-01]. Dostupné z: <http://microformats.org/wiki/microformats2>

MOLNÁR, Zdeněk, 2012. *Pokročilé metody vědecké práce*. 1. vyd. Zeleneč: Profess Consulting. ISBN 978-80-7259-064-3.

MONGODB INC., 2015. *Introduction to MongoDB* [online]. 2015. B.m.: MongoDB, Inc. Dostupné z: <https://docs.mongodb.org/getting-started/shell/introduction/>

MUSCIANO, Chuck a Bill KENNEDY, 2007. *HTML*. 6th ed. Sebastopol, CA: O'Reilly. ISBN 978-059-6527-327.

NIELSEN, Jakob, 1997. *The Use and Misuse of Focus Groups* [online] [vid. 2016-01-01]. Dostupné z: <https://www.nngroup.com/articles/focus-groups/>

NIKOLÍČ, Saša a Jurij ŠILC, 2016. Drupal 8 modules: Translation management tool and paragraphs. *Informatica (Slovenia)*. **40**(1), 145–152. ISSN 03505596.

NODE.JS FOUNDATION, 2015. *Node.js* [online]. 2015. B.m.: Node.js Foundation. Dostupné z: <https://nodejs.org/en/>

NURSEITOV, Nurzhan, Michael PAULSON, Randall REYNOLDS a Clemente IZURIETA, 2009. Comparison of JSON and XML Data Interchange Formats: A Case Study. *Scenario* [online]. **59715**, 1–3. Dostupné z: <http://www.cs.montana.edu/izurieta/pubs/caine2009.pdf>

OMG, 2015a. *ABOUT THE UNIFIED MODELING LANGUAGE SPECIFICATION VERSION 2.5* [online]. Dostupné z: <http://www.omg.org/spec/UML/>

OMG, 2015b. *OMG Unified Modeling Language (OMG UML)*. 2015. B.m.: OMG (Object Management Group).

PASQUALI, Sandro, 2013. *Mastering Node.js*. Birmingham: Packt Publishing. ISBN 978-1-78216-633-7.

- PHP GROUP, 2015. *What is PHP?* [online]. 2015. Dostupné z: <http://cz2.php.net/manual/en/intro-whatism.php>
- POWERMAPPER, 2017. *HTML 5 Version Statistics: HTML and XHTML versions commonly used* [online]. Dostupné z: <https://try.powermapper.com/Stats/HtmlVersions>
- RAGGETT, Dave, 1997. *HTML 3.2 Reference Specification* [online]. 1997. 14.-NaN.-199. vyd. Dostupné z: <http://www.w3.org/TR/REC-html32>
- RAGGETT, Dave, 1998. *Raggett on HTML 4*. 2nd ed. Reading, Mass.: Addison-Wesley. ISBN 02-011-7805-2.
- REITZ, Joan M, 2017. *ODLIS* [online]. 2017. Dostupné z: http://www.abc-clio.com/ODLIS/odlis_i.aspx
- SAHAY, Sundeep, 2003. Global software alliances : the challenge of 'standardization'. *Scandinavian Journal of Information Systems* [online]. **15**(1), 3–19. Dostupné z: <http://aisel.aisnet.org/sjis/vol15/iss1/11>
- SATCOUNTER, 2017. *Desktop vs Mobile vs Tablet Market Share Worldwide / StatCounter Global Stats* [online] [vid. 2018-01-13]. Dostupné z: <http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>
- SCHAFER, Steven M, 2009. *HTML, XHTML a CSS*. 1. vyd. Praha: Grada. ISBN 978-80-247-2850-6.
- SCHEMA.ORG, 2017. *About Schema.org* [online]. Dostupné z: <http://schema.org>
- SEADLE, Michael, 2006. Content management systems. *Library Hi Tech* [online]. **24**(issue 1), 5–7. ISSN 0737-8831. Dostupné z: doi:10.1108/07378830610652068
- SEEMAN, Mark, 2014. *SOLID: the next step is Functional* [online]. Dostupné z: <http://blog.ploeh.dk/2014/03/10/solid-the-next-step-is-functional/>
- SELWYN, Neil, 2011. „It's All about Standardisation" Exploring the Digital (Re)Configuration of School Management and Administration. *Cambridge Journal of Education* [online]. **41**(4), 473–488. ISSN 0305-764X. Dostupné z: doi:10.1080/0305764X.2011.625003
- SENGSTACKE, Peleke, 2016. JavaScript Transpilers: What They Are & Why We Need Them — Scotch. *scotch.io* [online]. Dostupné z: <https://scotch.io/tutorials/javascript-transpilers-what-they-are-why-we-need-them>
- SHANKLAND, Stephen, 2014. *HTML5 is done, but two groups still wrestle over Web's future* [online]. 2014. Dostupné z: <http://www.cnet.com/news/html5-is-done-but-two-groups-still-wrestle-over-webs-future/>

SPIESSER, J a L KITCHEN, 2004. Optimization of HTML automatically generated by WYSIWYG programs. In: *Thirteenth International World Wide Web Conference Proceedings, WWW2004* [online]. s. 355–364. Dostupné z: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-19944369446&partnerID=40&md5=476eaa75b8614f9482862279bb4062df>

STACK OVERFLOW, 2017. Stack Overflow Developer Survey 2017. *stackoverflow.com* [online]. Dostupné z: <https://insights.stackoverflow.com/survey/2017#technology>

STEFANOV, Stoyan a Kumar Chetan SHARMA, 2013. *Object-oriented JavaScript*. 2nd ed. Birmingham, UK: Packt Publishing. ISBN 978-1-84969-312-7.

SWENEY, Peter, 2016. The History of the Semantic Web is the Future of Intelligent Assistants. *Medium.com* [online]. Dostupné z: <https://medium.com/inventing-intelligent-machines/the-history-of-the-semantic-web-is-the-future-of-intelligent-assistants-da2ed50443be>

TVRDÍKOVÁ, M, 2001. Příprava společnosti na zavádění a inovace informačního systému. In: *Sborník konference Systémová integrace*. Praha: VŠE, s. 613–622.

U.S. DEPT. OF HEALTH AND HUMAN SERVICES, 2013. Focus Groups. *The Research-Based Web Design & Usability Guidelines, Enlarged/Expanded edition* [online]. B.m.: Department of Health and Human Services. Dostupné z: <https://www.usability.gov/how-to-and-tools/methods/focus-groups.html>

VAISH, Gaurav, 2013. *Getting started with NoSQL*. Birmingham: Packt Publishing. ISBN 978-1-84969-4-988.

VANÍČEK, Jiří, 2007. *Teoretické základy informatiky*. 1. vyd. Praha: Kernberg. ISBN 978-80-903962-4-1.

VOGELS, Rebecca, 2017. 7 Web Development Trends You Can Expect in 2018 - Usersnap. *USERSNAP* [online]. Dostupné z: <https://usersnap.com/blog/web-development-trends-2018/>

VYMĚTAL, Jan, Anna DIAČIKOVÁ a Miriam VÁCHOVÁ, 2005. *Informační a znalostní management v praxi*. Vyd. 1. Praha: LexisNexis CZ. ISBN 80-869-2001-1.

W3C, 1999. *HTML 4.01 Specification* [online]. 1999. Dostupné z: <http://www.w3.org/TR/html401/>

W3C, 2002. *XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)* [online]. 2002. W3C Recomm. Dostupné z: <http://www.w3.org/TR/xhtml1/>

W3C, 2004. *Architecture of the World Wide Web, Volume One* [online]. 2004. Dostupné z: <http://www.w3.org/TR/webarch/>

W3C, 2005. *Document Object Model (DOM)* [online]. 2005. Dostupné z: <http://www.w3.org/DOM/>

W3C, 2008. *Extensible Markup Language (XML) 1.0* [online]. 2008. W3C Recomm. Dostupné z: <http://www.w3.org/TR/xml/>

W3C, 2009. *Resource Description Framework (RDF)* [online] [vid. 2017-01-01]. Dostupné z: <http://www.w3.org/RDF/>

W3C, 2010. *XHTML™ 1.1 - Module-based XHTML - Second Edition* [online]. 2010. Dostupné z: <http://www.w3.org/TR/xhtml11/>

W3C, 2012. *Media Queries* [online] [vid. 2018-01-13]. Dostupné z: <https://www.w3.org/TR/css3-mediaqueries/>

W3C, 2014a. Common idioms without dedicated elements. *HTML5 A vocabulary and associated APIs for HTML and XHTML* [online]. Dostupné z: <https://www.w3.org/TR/html5/common-idioms.html#common-idioms>

W3C, 2014b. *HTML5 A vocabulary and associated APIs for HTML and XHTML* [online]. Dostupné z: <https://www.w3.org/TR/2014/REC-html5-20141028/>

W3C, 2014c. *JSON-LD 1.0* [online] [vid. 2017-01-01]. Dostupné z: <http://www.w3.org/TR/2014/REC-json-ld-20140116/>

W3C, 2015a. *FACTS ABOUT W3C* [online]. 2015. Dostupné z: <http://www.w3.org/Consortium/facts>

W3C, 2015b. *HTML+RDFa 1.1 - Second Edition* [online] [vid. 2017-01-01]. Dostupné z: <https://www.w3.org/TR/2015/REC-html-rdfa-20150317/>

W3C, 2015c. *SEMANTIC WEB* [online]. Dostupné z: <https://www.w3.org/standards/semanticweb/>

W3C, 2017a. *HTML 5.1 2nd Edition* [online] [vid. 2017-01-01]. Dostupné z: <https://www.w3.org/TR/html51/>

W3C, 2017b. *HTML 5.2* [online]. Dostupné z: <https://www.w3.org/TR/2017/REC-html52-20171214/>

W3C, 2017c. *HTML Microdata* [online] [vid. 2017-01-01]. Dostupné z: <https://www.w3.org/TR/microdata/>

W3SCHOOLS, 2015. *AJAX Introduction* [online]. Dostupné z: https://www.w3schools.com/xml/ajax_intro.asp

W3SCHOOLS, 2016. *CSS Syntax and Selectors* [online]. Dostupné z: https://www.w3schools.com/css/css_syntax.asp

W3SCHOOLS, 2017. *XML Introduction* [online] [vid. 2018-01-20]. Dostupné z: https://www.w3schools.com/xml/xml_what.asp

W3TECHS, 2018. *Usage Statistics of HTTP/2 for Websites, January 2018* [online]. Dostupné z: <https://w3techs.com/technologies/details/ce-http2/all/all>

WEB ACCESSIBILITY INICIATIVE, 2016. *WAI-ARIA Overview / Web Accessibility Initiative (WAI) / W3C* [online]. Dostupné z: <https://www.w3.org/WAI/intro/aria>

WEBAPPRATER, 2017. *Responsive Design vs. Native App vs. Web App - WebAppRater* [online] [vid. 2018-01-13]. Dostupné z: <https://webapprater.com/general/responsive-design-vs-native-app-vs-web-app.html>

WEINBERG, Randy S., 1991. Prototyping and the Systems Development Life Cycle. *Journal of Information Systems Management* [online]. **8**(2), 47–53. ISSN 0739-9014. Dostupné z: doi:10.1080/07399019108964983

WHATWG, 2018. *The Web Hypertext Application Technology Working Group (WHATWG)* [online]. 2018. Dostupné z: <https://html.spec.whatwg.org/>

WIKIMEDIA COMMONS, 2017. *File:CSS3 taxonomy and status by Sergey Mavrody.svg - Wikimedia Commons* [online]. Dostupné z: https://commons.wikimedia.org/wiki/File:CSS3_taxonomy_and_status_by_Sergey_Mavrody.svg

WRIGHT, A, 2017. *JSON Schema: A Media Type for Describing JSON Documents* [online]. 2017. B.m.: IETF. Dostupné z: <https://tools.ietf.org/html/draft-wright-json-schema-01>

10 Přílohy

10.1 Seznam obrázků

Obrázek č. 1 - Postup tvorby metodiky; zdroj: vlastní zpracování	8
Obrázek č. 2 - Znalostní pyramida; zdroj: (Chytková a Černý, 2012)	15
Obrázek č. 3 - Webové technologie - rozdělení client side, server side; zdroj: vlastní zpracování	18
Obrázek č. 4 - Rozložení verzí HTML na webových stránkách; zdroj: (PowerMapper, 2017)	19
Obrázek č. 5 - Typy elementů HTML5; zdroj: (W3C, 2017b)	22
Obrázek č. 6 - Moduly CSS; zdroj: (Wikimedia Commons, 2017)	24
Obrázek č. 7 - Syntaxe jazyka CSS; zdroj: (W3Schools, 2016)	25
Obrázek č. 8 - Statistika přístupů k webovým stránkám dle typu zařízení; zdroj: (Satcounter, 2017)	26
Obrázek č. 9 - Princip responzivního designu; zdroj: (WebAppRater, 2017)	27
Obrázek č. 10 - Struktura syntaxe JSON - objekt; zdroj: (JSON.org, 2018)	30
Obrázek č. 11 - Struktura syntaxe JSON - pole; zdroj: (JSON.org, 2018)	31
Obrázek č. 12 - JavaScript v modelu client-server; zdroj: (Goodman, 2010)	32
Obrázek č. 13 - Základní komunikace pomocí protokolu HTTP; zdroj: (Hock-Chuan, 2009)	36
Obrázek č. 14 - Statistiky užívání programovacího jazyka uživateli (programátory) webového fóra stackoverflow.com; zdroj: (Stack Overflow, 2017)	38
Obrázek č. 15 - Obsazení trhu redakčních systémů; zdroj: (Mening, 2017)	48

Obrázek č. 16 - Editace šablony polí obsahu; zdroj: vlastní zpracování	50
Obrázek č. 17 - Ukázka možností modulu Paragraphs; zdroj: vlastní zpracování... 51	51
Obrázek č. 18 - Základní editace obsahu v redakčním systému Joomla; zdroj: vlastní zpracování	52
Obrázek č. 19 - Editace šablony bloků obsahu; zdroj: vlastní zpracování	54
Obrázek č. 20 - Základní prototyp a princip oddělení WYS (What You See) a WYG (What You Get); zdroj: vlastní zpracování	63
Obrázek č. 21 - Struktura dat výchozího prototypu v NoSQL databázi Firebase; zdroj: vlastní zpracování	64
Obrázek č. 22 - Náhled vyvinutého prototypu - část 1/2; zdroj: vlastní zpracování	91
Obrázek č. 23 - Náhled vyvinutého prototypu - část 2/2; zdroj: vlastní zpracování	92

10.2 Seznam tabulek

Tabulka č. 1 - Historie verzí HTML; zdroj: vlastní zpracování.....	19
Tabulka č. 2 Pravidla pro nový content model na nejnižší úrovni ukládání obsahu	71
Tabulka č. 3 - České ekvivalenty klíčových slov z RFC2119	80
Tabulka č. 4 - Metodika WICM, atributy třídy Article; zdroj: vlastní zpracování.....	82
Tabulka č. 5 - Metodika WICM, atributy třídy Block; zdroj: vlastní zpracování.....	83
Tabulka č. 6 - Metodika WICM, atributy třídy Block Group; zdroj: vlastní zpracování	83

Tabulka č. 7 - Metodika WICM, atributy třídy List; zdroj: vlastní zpracování.....	84
Tabulka č. 8 - Metodika WICM, atributy třídy List Item; zdroj: vlastní zpracování	84
Tabulka č. 9 - Metodika WICM, atributy třídy Table Cell; zdroj: vlastní zpracování	85
Tabulka č. 10 - Metodika WICM, atributy třídy Simple content; zdroj: vlastní zpracování	85
Tabulka č. 11 - Metodika WICM, atributy třídy Embedded content; zdroj: vlastní zpracování	85
Tabulka č. 12 - Metodika WICM, atributy třídy HTML; zdroj: vlastní zpracování ...	86
Tabulka č. 13 - Metodika WICM, atributy třídy Comment; zdroj: vlastní zpracování	86

10.3 Seznam zdrojových kódů

Zdrojový kód č. 1 - Ukázka zápisu metadatového formátu Microdata do HTML; zdroj: vlastní zpracování	42
Zdrojový kód č. 2 Základní HTML šablona s povinnými elementy	75
Zdrojový kód č. 3 - Základní HTML šablona doplněná o metadata ve formátu Microdata.....	78
Zdrojový kód č. 4 Základní JSON Schema	89

10.4 Seznam diagramů

Diagram č. 1 - Základní struktura obsahu – Class diagram UML, doménový model; zdroj: vlastní zpracování	66
--	----

Diagram č. 2 - Model ukládaného obsahu - Class diagram UML, doménový model; zdroj: vlastní zpracování	72
Diagram č. 3 - Model uložení obsahu doplněný o podporu metadatového popisu; zdroj: vlastní zpracování	77
Diagram č. 4 - Model uložení obsahu; zdroj: vlastní zpracování	81

10.5 Exportní formát JSON Schema

<http://kit.pef.czu.cz/JSONschemas/wicm.json>

```
{
  "$id": "http://kit.pef.czu.cz/JSONschemas/wicm.json",
  "$schema": "http://json-schema.org/draft-06/schema#",
  "type": "object",
  "properties": {
    "headline": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "dateCreated": {
      "type": "string",
      "format": "date-time"
    },
    "dateUpdated": {
      "type": "string",
      "format": "date-time"
    },
    "datePublished": {
      "type": "string",
      "format": "date-time"
    },
    "associatedMedia": {
      "type": "object",
      "$ref": "http://kit.pef.czu.cz/JSONschemas/embedded_schema.json"
    },
    "articleSections": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
```

```

"blocks": {
  "type": "array",
  "items": {
    "type": "object",
    "$ref": "http://kit.pef.czu.cz/JSONschemas/block_schema.json"
  }
},
"blockGroups": {
  "type": "array",
  "items": {
    "type": "object",
    "$ref": "http://kit.pef.czu.cz/JSONschemas/blockGroup_schema.json"
  }
},
"schemaType": {
  "type": "string"
},
"schemaProperties": {
  "type": "array",
  "minItems": 0,
  "items": {
    "type": "object",
    "properties": {
      "propertyName": {
        "type": "string"
      },
      "propertyValue": {
        "type": "string"
      }
    }
  }
},
"required": [
  "headline",
  "dateCreated"
]
}

```

http://kit.pef.czu.cz/JSONschemas/block_schema.json

```

{
  "$id": "http://kit.pef.czu.cz/JSONschemas/block_schema.json",
  "$schema": "http://json-schema.org/draft-06/schema#",
  "type": "object",
  "properties": {
    "blockID": {
      "type": "integer"
    },
    "position": {
      "type": "integer"
    }
  },
}

```

```

"heading": {
  "type": "string"
},
"schemaType": {
  "type": "string"
},
"schemaProperty": {
  "type": "string"
},
"schemaProperties": {
  "type": "array",
  "minItems": 0,
  "items": {
    "type": "object",
    "properties": {
      "propertyName": {
        "type": "string"
      },
      "propertyValue": {
        "type": "string"
      }
    }
  }
},
"paragraph": {
  "type": "object",
  "$ref": "http://kit.pef.czu.cz/JSONschemas/simpleContent_schema.json"
},
"list": {
  "type": "object",
  "$ref": "http://kit.pef.czu.cz/JSONschemas/list_schema.json"
},
"table": {
  "type": "object",
  "$ref": "http://kit.pef.czu.cz/JSONschemas/table_schema.json"
},
"simpleContent": {
  "type": "object",
  "$ref": "http://kit.pef.czu.cz/JSONschemas/simpleContent_schema.json"
},
"HTML": {
  "type": "object",
  "$ref": "#/definitions/HTML"
},
"embedded": {
  "type": "object",
  "$ref": "http://kit.pef.czu.cz/JSONschemas/embedded_schema.json"
}
},
"required": [
  "blockID"
],
"definitions": {

```

```
"HTML": {
  "type": "object",
  "properties": {
    "content": {
      "type": "string"
    }
  }
}
```

http://kit.pef.czu.cz/JSONschemas/blockGroup_schema.json

```
{
  "$id": "http://kit.pef.czu.cz/JSONschemas/blockGroup_schema.json",
  "$schema": "http://json-schema.org/draft-06/schema#",
  "type": "object",
  "properties": {
    "blockGroupID": {
      "type": "integer"
    },
    "position": {
      "type": "integer"
    },
    "heading": {
      "type": "string"
    },
    "schemaType": {
      "type": "string"
    },
    "template": {
      "type": "string"
    },
    "schemaProperties": {
      "type": "array",
      "minItems": 0,
      "items": {
        "type": "object",
        "properties": {
          "propertyName": {
            "type": "string"
          },
          "propertyValue": {
            "type": "string"
          }
        }
      }
    },
    "blocks": {
      "type": "array",
      "items": {
        "type": "object",
```

```

    "$ref": "http://kit.pef.czu.cz/JSONschemas/block_schema.json"
  },
  "blockGroups": {
    "type": "array",
    "items": {
      "type": "object",
      "$ref": "http://kit.pef.czu.cz/JSONschemas/blockGroup_schema.json"
    }
  }
},
"required": [
  "blockID"
]
}

```

http://kit.pef.czu.cz/JSONschemas/list_schema.json

```

{
  "$id": "http://kit.pef.czu.cz/JSONschemas/list_schema.json",
  "$schema": "http://json-schema.org/draft-06/schema#",
  "type": "object",
  "properties": {
    "listType": {
      "type": "string",
      "oneOf": [
        {
          "format": "ul"
        },
        {
          "format": "ol"
        },
        {
          "format": "dl"
        }
      ]
    },
    "listItems": {
      "type": "array",
      "items": {
        "type": "object",
        "oneOf": [
          {
            "$ref": "http://kit.pef.czu.cz/JSONschemas/simpleContent_schema.json"
          },
          {
            "$ref": "http://kit.pef.czu.cz/JSONschemas/list_schema.json"
          }
        ]
      }
    }
  }
}

```

```
}  
}
```

http://kit.pef.czu.cz/JSONschemas/listItem_schema.json

```
{  
  "$id": "http://kit.pef.czu.cz/JSONschemas/listItem_schema.json",  
  "$schema": "http://json-schema.org/draft-06/schema#",  
  "type": "object",  
  "properties": {  
    "itemType": {  
      "type": "string",  
      "oneOf": [  
        {  
          "format": "dt"  
        },  
        {  
          "format": "dd"  
        },  
        {  
          "format": "li"  
        }  
      ]  
    },  
    "simpleContent": {  
      "$ref": "http://kit.pef.czu.cz/JSONschemas/simpleContent_schema.json"  
    }  
  }  
}
```

http://kit.pef.czu.cz/JSONschemas/table_schema.json

```
{  
  "$id": "http://kit.pef.czu.cz/JSONschemas/table_schema.json",  
  "$schema": "http://json-schema.org/draft-06/schema#",  
  "type": "object",  
  "properties": {  
    "Thead": {  
      "type": "array",  
      "items": {  
        "type": "array",  
        "$ref": "#/definitions/table_row"  
      }  
    },  
    "Tbody": {  
      "type": "array",  
      "items": {  
        "type": "array",  
        "$ref": "#/definitions/table_row"  
      }  
    }  
  }  
}
```

```

    },
    "tfoot": {
      "type": "array",
      "items": {
        "type": "array",
        "$ref": "#/definitions/table_row"
      }
    },
    "caption": {
      "type": "object",
      "$ref": "simpleContent_schema.json"
    }
  },
  "required": [
    "tbody"
  ],
  "definitions": {
    "tbody_row": {
      "type": "array",
      "items": {
        "type": "object",
        "$ref": "#/definitions/table_cell"
      }
    },
    "table_cell": {
      "type": "object",
      "properties": {
        "isHead": {
          "type": "boolean"
        },
        "colspan": {
          "type": "number"
        },
        "rowspan": {
          "type": "number"
        },
        "simpleContent": {
          "type": "object",
          "$ref": "http://kit.pef.czu.cz/JSONschemas/simpleContent_schema.json"
        }
      }
    }
  }
}

```

http://kit.pef.czu.cz/JSONschemas/simpleContent_schema.json

```

{
  "$id": "http://kit.pef.czu.cz/JSONschemas/simpleContent_schema",
  "$schema": "http://json-schema.org/draft-06/schema#",
  "type": "object",

```

```

"properties": {
  "content": {
    "type": "string"
  },
  "schemaProperty": {
    "type": ["string", "null"]
  }
},
"required": [
  "content"
]
}

```

http://kit.pef.czu.cz/JSONschemas/embedded_schema.json

```

{
  "$id": "http://kit.pef.czu.cz/JSONschemas/embedded_schema.json",
  "$schema": "http://json-schema.org/draft-06/schema#",
  "type": "object",
  "properties": {
    "attributes": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {"type": "string"},
          "value": {"type": "string"}
        }
      }
    },
    "element": {
      "type": "string"
    },
    "schemaProperty": {
      "type": ["string", "null"]
    }
  },
  "required": [
    "attributes"
  ]
}

```


10.6 Publikační činnost autora

Články

MASNER, J. – VANĚK, J. – STOČES, M. Spatial Data Monitoring and Mobile Applications – Comparison of Methods for Parsing JSON in Android Operating System. *AGRIS on-line Papers in Economics and Informatics*, 2014, roč. VI, č. 1, s. 37-46. ISSN: 1804-1930.

JAROLÍMEK, J. – MASNER, J. – ULMAN, M. – DVOŘÁK, S. Cloven-hoofed animals spatial activity evaluation methods in Doupov Mountains in the Czech Republic. *AGRIS on-line Papers in Economics and Informatics*, 2012, roč. IV, č. 3, s. 41-48. ISSN: 1804-1930.

VANĚK, J. – BROŽOVÁ, I. – MASNER, J. – ŠIMEK, P. – VOGELTANZOVÁ, T. Information Support of Regions – Organic Farming. *AGRIS on-line Papers in Economics and Informatics*, 2013, roč. V, č. 3, s. 97-104. ISSN: 1804-1930.

ŠIMEK, P. – STOČES, M. – VANĚK, J. – JAROLÍMEK, J. – MASNER, J. – HRBEK, I. Using of Automatic Metadata Providing. *AGRIS on-line Papers in Economics and Informatics*, 2013, roč. V, č. 4, s. 189-197. ISSN: 1804-1930.

ŠIMEK, P. – JAROLÍMEK, J. – MASNER, J. Cross-Platform User Interface of a Web Application in Agrarian Sector. *AGRIS on-line Papers in Economics and Informatics*, 2014, roč. VI, č. 4, s. 155-160. ISSN: 1804-1930.

STOČES, M. – MASNER, J. – JAROLÍMEK, J. Mitigation of Social Exclusion in Regions and Rural Areas – E-learning with Focus on Content Creation and Evaluation. *AGRIS on-line Papers in Economics and Informatics*, 2015, roč. 7, č. 4, s. 143-150. ISSN: 1804-1930.

MASNER, J. – ŠIMEK, P. – JAROLÍMEK, J. – HRBEK, I. Mobile Applications for Agricultural Online Portals – Cross-platform or Native Development. *AGRIS on-line Papers in Economics and Informatics*, 2015, roč. 7, č. 2, s. 47-54. ISSN: 1804-1930.

STOČES, M. – VANĚK, J. – MASNER, J. – PAVLÍK, J. Internet of Things (IoT) in Agriculture – Selected Aspects. *AGRIS on-line Papers in Economics and Informatics*, 2016, roč. 8, č. 1, s. 83-88. ISSN: 1804-1930.

BENDA, P. – ŠIMEK, P. – MASNER, J. – VANĚK, J. Analysis of eAGRI web portal ergonomics and presentation of information in terms of the general public. *AGRIS on-line Papers in Economics and Informatics*, 2017, roč. 9, č. 4, s. 3-13. ISSN: 1804-1930.

JAROLÍMEK, J. – VANĚK, J. – JEŽEK, M. – MASNER, J. – STOČES, M. The telemetric tracking of wild boar as a tool for field crops damage limitation. *Plant, Soil and Environment*, 2014, roč. 60, č. 9, s. 418-425. ISSN: 1214-1178.

Články ve sborníku

MASNER, J. – STOČES, M. Web application for Collection and visualization of positional data – Wild boars and deer monitoring in Czech Republic. In *Agrarian Perspectives – Proceedings of the 22th Scientific Conference: Development trends in agribusiness*

17.09.2013, *Czech University of Life Sciences Prague, CR*. Prague, CR: CULS Prague, Faculty of Economics and Management, 2013. s. 141-147.

MASNER, J. – STOČES, M. Sběr a zobrazení pozičních dat pomocí webové aplikace. In *Think Together 2014, Doktorská vědecká konference 03.02.2014, Praha*. Praha: Provozně ekonomická fakulta, Česká zemědělská univerzita v Praze, 2014. s. 1-1.

STOČES, M. – MASNER, J. – JAROLÍMEK, J. – ŠIMEK, P. – VANĚK, J. – ULMAN, M. Cross-platform user interface of e-learning applications. In *Proceedings of the 11th International Conference on Mobile Learning 2015, ML 2015 14.03.2015, Madeira*. Madeira; Portugal: IADIS, 2015. s. 135-138.

VANĚK, J. – ŠIMEK, P. – STOČES, M. – ULMAN, M. – MASNER, J. – JAROLÍMEK, J. DIGITAL DIVIDE AND ICT DEVELOPMENT IN RURAL AREAS IN THE CZECH REPUBLIC. In *PROCEEDINGS OF THE 13th INTERNATIONAL CONFERENCE e-Society 2015 14.03.2015, Funchal, Madeira - Portugal*. Portugal: IADIS Press, 2015. s. 331-334.

ULMAN, M. – JAROLÍMEK, J. – ŠIMEK, P. – MASNER, J. – VANĚK, J. – STOČES, M. EGOVERNMENT IN THE CZECH REPUBLIC – DO DATA RUN OR CITIZENS?. In *PROCEEDINGS OF THE 13th INTERNATIONAL CONFERENCE e-Society 2015 14.03.2015, Funchal, Madeira*. : IADIS Press, 2015. s. 255-259.

MASNER, J. – VANĚK, J. – JAROLÍMEK, J. – OČENÁŠEK, V. Markup Languages Support for Content Management of Agricultural Portals. In *Proceedings of the 7th International Conference on Information and Communication Technologies in Agriculture, Food and Environment 17.09.2015, Kavala, Greece*. : CEUR-WS, 2015. s. 976-982.

ŠMEJKALOVÁ, M. – MASNER, J. – JAROLÍMEK, J. Methodology of usability testing of the next generation of editors for agricultural web portals. In *Agrarian Perspectives XXIV. – Global Agribusiness and Rural Economy 16.09.2015, Prague*. Praha: Czech University of Life Sciences Prague, 2015. s. 456-462.

ŠIMEK, P. – STOČES, M. – VANĚK, J. – MASNER, J. Mobile accessibility of information sources in the areas of agriculture, forestry, water management, food industry and rural development. In *Agrarian Perspectives XXIV. – Global Agribusiness and Rural Economy 16.09.2015, Prague*. Praha: Czech University of Life Sciences Prague, Faculty of Economics and Management, 2015. s. 440-446.

STOČES, M. – VANĚK, J. – MASNER, J. – JAROLÍMEK, J. Mobile application development options for news and information portals. In *Future Communication, Information and Computer Science 22.05.2014, Beijing*. Beijing: CRC press Taylor & Francis group, 2015. s. 111-114.

ULMAN, M. – ŠILEROVÁ, E. – MASNER, J. – STOČES, M. – ŠIMEK, P. – BENDA, P. Open data in regions from the users' perspective: an analytical study. In *Proceedings of the 7th International Conference on Information and Communication Technologies in Agriculture, Food and Environment 17.09.2015, Kavala, Greece*. Kavala, Greece: , 2015. s. 902-911.

MASNER, J. – STOČES, M. – VANĚK, J. Spatial and activity data-collection and visualization in WWW environment . In *Future Communication, Information and*

Computer Science 22.05.2014, Beijing. Beijing: CRC Press Taylor & Francis Group, 2015. s. 97-100.

JAROLÍMEK, J. – ULMAN, M. – VANĚK, J. – MASNER, J. – STOČES, M. Use of ICT in education for mitigation of social exclusion. In *Future Communication, Information and Computer Science 22.05.2014, Beijing, China*. : CRC Press, 2015. s. 1-10.

VANĚK, J. – OČENÁŠEK, V. – STOČES, M. – MASNER, J. Adoption of modern ICT in regional perspective – situation in the Czech Republic. In *GLOBALIZATION AND ITS SOCIO-ECONOMIC CONSEQUENCES 05.10.2016, Žilina, SK*. Žilina, SK: GEORG, Bajzova 11, 010 01 Zilina, Slovak Republic, 2016. s. 2291-2298.

MASNER, J. – VANĚK, J. – JAROLÍMEK, J. Prototype of a Content Creation and Updating Application Module for Agrarian Sector and Regional Development. In *AGRARIAN PERSPECTIVES XXV: Global and European Challenges for Food Production, Agribusiness and the Rural Economy 13.09.2016, Praha*. Praha: CULS Prague, 2016. s. 1-2.

PAVLÍK, J. – MASNER, J. – RAJTR, J. Data processing methods for information system testing in agrarian sector. In *Proceedings of the 10th IADIS International Conference on Information Systems 2017, IS 2017 10.04.2017, Budapešť*. neuvvedeno: IADIS - International Association for the Development of the Information Society, 2017. s. 207-210.

BENDA, P. – MASNER, J. – VANĚK, J. – RAJTR, J. Ergonomics of the eAGRI agrarian portal. In *Agrarian perspectives XXVI. – COMPETITIVENESS OF EUROPEAN AGRICULTURE AND FOOD SECTORS 13.09.2017, Prague*. Prague: Czech University of Life Sciences Prague, 2017. s. 8-14.

OČENÁŠEK, V. – MASNER, J. – VANĚK, J. – ŠILEROVÁ, E. – PAVLÍK, J. Evaluation of Farmer's E-shops. In *Proceedings of the 8th International Conference on Information and Communication Technologies in Agriculture, Food and Environment (HAICTA 2017) 21.09.2017, Chania, Crete Island, Greece*. : CEUR-WS, 2017. s. 224-231.

STOČES, M. – PAVLÍK, J. – MASNER, J. – JUNGWIRTH, M. – HAVLÍČEK, Z. Monitoring visualization of the animal movement in buildings. In *AGRARIAN PERSPECTIVES XXVI. COMPETITIVENESS OF EUROPEAN AGRICULTURE AND FOOD SECTORS 13.09.2017, Prague, Czech Republic*. Prague, Czech Republic: Czech University of Life Sciences Prague, 2017. s. 0-0.

MASNER, J. – JAROLÍMEK, J. – VANĚK, J. – ŠIMEK, P. – ŠILEROVÁ, E.; idPublikace = 75414; Název: User-Technological Index of Precision Agriculture: Methods of Collecting data-- Neexistuje podtyp publikace --

Software

MASNER, J. – JAROLÍMEK, J. – ULMAN, M. – ŠIMEK, P. – STOČES, M. – VANĚK, J. GPS Trace Route 1.0, GPS Trace Route 1.0, Positional data, GPS, polygon, quick hull, 2012, GA - Zemědělská ekonomie, GPS Trace Route 1.0, www.agris.cz/kit/tr, viz lokalizace; kontakt: Ing. Pavel Šimek, Ph.D., Katedra informačních technologií, PEF ČZU v Praze, Kamýcká 129 165 21 Praha 6 - Suchbát, +420 224 382 050, [simek\(at\)pef.czu.cz](mailto:simek(at)pef.czu.cz), Nové SW řešení: – nová kvalita, široká použitelnost, moderní řešení, Katedra informačních

technologií PEF ČZU v Praze, 60460709, CZ - Česká republika, A - K využití výsledku jiným subjektem je vždy nutné nabytí licence, N - Nevyžaduje se

JAROLÍMEK, J. – MASNER, J. – STOČES, M. – ULMAN, M. – ŠIMEK, P. – VANĚK, J. Ground Station Harvester (GSH 1.0), Ground Station Harvester (GSH 1.0), GPS, positional data, ground station, database, SW, 2012, GA - Zemědělská ekonomie, GSH 1.0, www.agris.cz/kit/gsh, viz lokalizace; kontakt: Ing. Pavel Šimek, Ph.D., Katedra informačních technologií, PEF ČZU v Praze, Kamýcká 129 165 21 Praha 6 - Suchdol, +420 224 382 050, [simek\(at\)pef.czu.cz](mailto:simek(at)pef.czu.cz), Nové SW řešení: – nová kvalita, široká použitelnost, moderní řešení., Katedra informačních technologií PEF ČZU v Praze, 60460709, CZ - Česká republika, A - K využití výsledku jiným subjektem je vždy nutné nabytí licence, N - Nevyžaduje se

MASNER, J. – ŠIMEK, P. – STOČES, M. – VANĚK, J. – JAROLÍMEK, J. Nativní mobilní aplikace Agrárního WWW portálu AGRIS pro platformu Android, Agrarian WWW portal AGRIS native mobile application for Android platform, agris, android, aplikace, google play, 2015, JC - Počítačový hardware a software, Agris mobile v. 1.0, <http://www.agris.cz/kit/agm/>, viz lokalizace; kontakt: Ing. Pavel Šimek, Ph.D., Katedra informačních technologií, PEF ČZU v Praze, Kamýcká 129 165 21 Praha 6 - Suchdol, +420 224 382 050, [simek\(at\)pef.czu.cz](mailto:simek(at)pef.czu.cz), Nové SW řešení: – nová kvalita, široká použitelnost, moderní řešení., Katedra informačních technologií PEF ČZU v Praze, 60460709, CZ - Česká republika, N - Využití výsledku jiným subjektem je možné bez nabytí licence (výsledek není licencován), N - Nevyžaduje se

STOČES, M. – JAROLÍMEK, J. – VANĚK, J. – ŠIMEK, P. – MASNER, J. Click Tracking Data Collector for UX, Click Tracking Data Collector for UX, web browser; tracking; user experience, 2016, JC - Počítačový hardware a software, CTDCUX 1.0, <http://www.agris.cz/kit/ctdcux>, viz lokalizace; kontakt: Ing. Pavel Šimek, Ph.D., Katedra informačních technologií, PEF ČZU v Praze, Kamýcká 129 165 21 Praha 6 - Suchdol, +420 224 382 050, [simek\(at\)pef.czu.cz](mailto:simek(at)pef.czu.cz), Nové SW řešení: – nová kvalita, široká použitelnost, moderní řešení., Katedra informačních technologií PEF ČZU v Praze, 60460709, CZ - Česká republika, N - Využití výsledku jiným subjektem je možné bez nabytí licence (výsledek není licencován), N - Nevyžaduje se

MASNER, J. – ŠIMEK, P. – VANĚK, J. – JAROLÍMEK, J. – STOČES, M. Electronic learning mobile application, Electronic learning mobile application, iOS; mobile application; e-learning; learning materials, 2016, JC - Počítačový hardware a software, ELMA 1.0, <http://www.agris.cz/kit/elma>, viz lokalizace; kontakt: Ing. Pavel Šimek, Ph.D., Katedra informačních technologií, PEF ČZU v Praze, Kamýcká 129 165 21 Praha 6 - Suchdol, +420 224 382 050, [simek\(at\)pef.czu.cz](mailto:simek(at)pef.czu.cz), Nové SW řešení: – nová kvalita, široká použitelnost, moderní řešení., Katedra informačních technologií PEF ČZU v Praze, 60460709, CZ - Česká republika, N - Využití výsledku jiným subjektem je možné bez nabytí licence (výsledek není licencován), N - Nevyžaduje se

PAVLÍK, J. – JUNEK, P. – VANĚK, J. – JAROLÍMEK, J. – ŠIMEK, P. – MASNER, J. – STOČES, M. – **GALLIE, M. Video Event Analyser for UX, Video Event Analyser for UX, UX; event; video; analyzer, 2016, JC - Počítačový hardware a software, VEAUX 1.0, <http://www.agris.cz/kit/veaux>, viz lokalizace; kontakt: Ing. Pavel Šimek, Ph.D., Katedra informačních technologií, PEF ČZU v Praze, Kamýcká 129 165 21 Praha 6 -

Suchdol, +420 224 382 050, simek(at)pef.czu.cz, Nové SW řešení: – nová kvalita, široká použitelnost, moderní řešení, Katedra informačních technologií PEF ČZU v Praze, 60460709, CZ - Česká republika, N - Využití výsledku jiným subjektem je možné bez nabytí licence (výsledek není licencován), N - Nevyžaduje se