

Univerzita Palackého v Olomouci
Přírodovědecká fakulta
Katedra geoinformatiky

Adéla OTEVŘELOVÁ

**EYE TRACKING VYHODNOCENÍ SALIENCE
A INTERAKTIVNÍ PRÁCE VE VIZUÁLNÍCH
PROGRAMOVACÍCH JAZYCÍCH**

Diplomová práce

Vedoucí práce: Ing. Zdena DOBEŠOVÁ, Ph.D.

Olomouc 2014

Čestné prohlášení

Prohlašuji, že jsem diplomovou práci magisterského studia oboru Geoinformatika vypracovala samostatně pod vedením Ing. Zdeny DOBEŠOVÉ, Ph.D.

Všechny použité materiály a zdroje jsou citovány s ohledem na vědeckou etiku, autorská práva a zákony na ochranu duševního vlastnictví.

Všechna poskytnutá i vytvořená digitální data nebudu bez souhlasu školy poskytovat.

V Českém Těšíně dne 22. dubna 2014

Adéla OTEVŘELOVÁ

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Adéla OTEVŘELOVÁ**
Osobní číslo: **R120616**
Studijní program: **N1301 Geografie**
Studijní obor: **Geoinformatika**
Název tématu: **Eye tracking vyhodnocení salience a interaktivní práce ve vizuálních programovacích jazycích**
Zadávající katedra: **Katedra geoinformatiky**

Z á s a d y p r o v y p r a c o v á n í :

Studentka zpracuje rešerši použití vizuální salience pro rozhraní vizuálních programovacích jazyků v GIS. Následně provede srovnání vizuální salience a výsledků eye-tracking testování pro vizuální programovací jazyky používaných v GIS. Dále připraví a realizuje eye-tracking testy pro zhodnocení možností interaktivní práce ve VPL.

Studentka vyplní údaje o všech datových sadách, které vytvořil nebo získal v rámci práce, do Metainformačního systému katedry geoinformatiky a současně zálohu údajů ve formě validovaného XML souboru. Celá práce (text, přílohy, výstupy, zdrojová a vytvořená data, XML soubor) se odevzdá v digitální podobě na CD (DVD) a text práce s vybranými přílohami bude odevzdán ve dvou svázaných výtiscích na sekretariát katedry. O magisterské práci studentka vytvoří webovou stránku v souladu s pravidly dostupnými na stránkách katedry. Práce bude zpracována podle zásad dle Voženílek (2002) a závazné šablony pro diplomové práce na KGI. Na závěr diplomové práce připojí studentka jednostránkové resumé v anglickém jazyce.

Rozsah grafických prací: **dle potřeby**
Rozsah pracovní zprávy: **max. 50 stran**
Forma zpracování diplomové práce: **tištěná**
Seznam odborné literatury:

Holmquist, K., Nyström, M., Andersson, R., Dewhurst, R., Halszka, J. & Van De Weijer, J. (2011). *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 560 pp.
Kudělka, V. (2013): *Srovnání vizuálního programování v GIS produktech podle kognitivních dimenzí*, bakalářská práce, Katedra geoinformatiky, Přírodovědecká fakulta, Univerzita Palackého, Olomouc.
Itti L. (2007): *Visual Saliency*,
http://www.scholarpedia.org/article/Visual_saliency
Zieglerová, J. (2012): *Vybrané aspekty uživatelského vnímání kartografických produktů*, bakalářská práce, Katedra geoinformatiky, Přírodovědecká fakulta, Univerzita Palackého, Olomouc.

Vedoucí diplomové práce: **Ing. Zdena Dobešová, Ph.D.**
Katedra geoinformatiky

Datum zadání diplomové práce: **12. prosince 2012**
Termín odevzdání diplomové práce: **23. dubna 2014**

UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA GEOINFORMATIKY
17. listopadu 50, 771 46 Olomouc
-1-

Prof. RNDr. Juraj Ševčík, Ph.D.
děkan

L.S.



Prof. RNDr. Vít Voženílek, CSc.
vedoucí katedry

V Olomouci dne 12. prosince 2012

OBSAH

ÚVOD	6
1 CÍLE PRÁCE	7
2 POUŽITÉ METODY A POSTUPY ZPRACOVÁNÍ	8
2.1 Použitá data	8
2.2 Použité programy	8
2.3 Postup zpracování	9
3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY	10
3.1 Salience	10
4 TESTOVÁNÍ SALIENCE METODOU EYE-TRACKING	14
4.1 Struktura experimentů	15
5 HODNOCENÍ VIZUÁLNÍ SALIENCE	17
5.1 OGAMA (Saliency Module)	17
5.2 Výpočet salience pro diagramy datových toků	20
5.3 Srovnání výsledků eye-tracking testování a vypočítané salience	23
5.3.1 Model Maker	23
5.3.2 Macro Modeler	25
5.3.3 ModelBuilder	27
5.3.4 Workflow Desinger	28
5.3.5 Shrnutí	30
5.4 Srovnání výsledků eye-tracking testování podle zkušeností s VPL	30
6 TESTOVÁNÍ INTERAKTIVNÍ PRÁCE METODOU EYE-TRACKING	32
6.1 Struktura experimentů	32
6.1.1 První testování	33
6.1.2 Druhé testování	34
6.2 Průběh testování	34
6.3 Hodnocení interaktivní práce	35
6.3.1 První testování	35
6.3.2 Druhé testování	43
6.3.3 Shrnutí	49
7 VÝSLEDKY	53
8 DISKUZE	54
9 ZÁVĚR	56
POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE	
SUMMARY	
PŘÍLOHY	

ÚVOD

Prostřednictvím vizuálních programovacích jazyků lze jednoduše programovat s použitím grafických prvků, které se propojují pomocí spojovacích šipek. Vizuální programování tedy spočívá ve tvorbě diagramů datových toků. Uživatelé mají možnost programovat bez znalostí syntaxe programovacích jazyků. U vizuálních programovacích jazyků hraje důležitou roli podoba jednotlivých grafických prvků, které poutají vizuální pozornost uživatelů.

Diplomová práce se zabývá vyhodnocením vizuální salience a interaktivní práce ve vizuálních programovacích jazycích. Srovnání generované salience a výsledků testování salience metodou eye-tracking by mělo poukázat na to, jestli podoba jednotlivých prvků v diagramech datových toků přitahuje pozornost uživatelů. Práce také řeší, jak důležité je uspořádání jednotlivých prvků v diagramech.

Zhodnocení interaktivní práce ve vybraném vizuálním programovacím jazyce by mělo ukázat, jak je přívětivé rozhraní dané aplikace pro uživatele. Dále v čem jsou nedostatky, jak na straně programu, tak uživatelů.

1 CÍLE PRÁCE

Cílem diplomové práce je eye-tracking testování a následné vyhodnocení vizuální salience. Dále zhodnocení interaktivní práce ve vizuálních programovacích jazycích. Zhotovení rešerše problematiky vizuální salience.

V praktické části práce bylo cílem vytvořit diagramy datových toků ve čtyřech vybraných vizuálních programovacích jazycích pro následné hodnocení vizuální salience. Poté připravit eye-tracking experimenty pro testování vizuální salience. Nakonec porovnat vizuální salience vypočítanou ve zvoleného programu s výsledky testování salience metodou eye-tracking.

Dále bylo cílem práce připravit eye-tracking experimenty pro testování interaktivní práce ve vybraném vizuálním programovacím jazyce. Sestavit zadání jednotlivých úkolů a následně vytvořit experimenty. Poté realizovat a vyhodnotit eye-tracking testování interaktivní práce ve vybraném vizuálním programovacím jazyce.

2 POUŽITÉ METODY A POSTUPY ZPRACOVÁNÍ

Zpracování diplomové práce probíhalo v následujících krocích. První fází bylo studium dostupné literatury dané problematiky a tvorba rešerše. Praktická část byla rozdělena na dvě hlavní části, první se zabývala vizuální salienčí a druhá interaktivní prací. Hlavní náplní této práce bylo testování metodou eye-tracking. Nejdříve probíhala tvorba diagramů a příprava eye-tracking testů pro testování vizuální salience ve vizuálních programovacích jazycích používaných v GIS. Následovalo vyhodnocení a srovnání výsledků eye-tracking testů s vizuální salienčí vypočítanou v programu OGAMA. Dalším krokem byl návrh a realizace eye-tracking testů pro zhodnocení interaktivní práce ve vizuálních programovacích jazycích. Poté proběhlo vyhodnocení eye-tracking testů interaktivní práce.

2.1 Použitá data

Při přípravě eye-tracking testů byly vytvořeny sady diagramů ve vybraných VPL (Visual Programming Language) komponentách. Snahou bylo vytvořit funkční diagramy, proto bylo potřeba použít vstupní data. Při tvorbě diagramů v programech primárně určených pro práci s rastrovými daty (ERDAS Imagine a Idrisi) byly využity testovací data, které jsou součástí instalačních balíčků. Při práci v programu ArcGIS for Desktop Advanced a Autodesk AutoCAD Map 3D byla použita digitální geografická databáze České republiky ArcČR 500 v3.1, případně ArcČR 500 v2.0.

2.2 Použité programy

Pro hodnocení vizuální salience byly vybrány čtyři aplikace. Prvním programem byl ERDAS Imagine 2013 s VPL komponentou Model Maker od firmy Integraph, která je dceřinnou společností švédské společnosti Hexagon AB. Od této verze obsahuje ERDAS Imagine navíc také VPL komponentu Spatial Model Editor. Další VPL komponentou byl Macro Modeler zahrnutý v programu Idrisi 17.0 The Selva Edition, který byl vyvinut Clark Labs na americké výzkumné univerzitě Clark University. Od americké společnosti Esri byla použita aplikace ArcGIS 10.1 for Desktop Advanced s VPL komponentou ModelBuilder, která byla také vybrána pro hodnocení interaktivní práce. Poslední testovaným programem byl Autodesk AutoCAD Map 3D 2014 od americké firmy Autodesk, Inc. s VPL komponentou Workflow Designer. Grafické úpravy jednotlivých diagramů byly provedeny v open source grafickém editoru GIMP.

Testování metodou eye-tracking probíhalo v eye-tracking laboratoři na Katedře geoinformatiky Univerzity Palackého v Olomouci pomocí přístroje RED 250 od společnosti SMI (SensoMotoric Instruments). K přípravě testů, průběhu testování a exportu naměřených dat byla použita sada programů SMI Experiment Suite™ 360°, která obsahuje SMI Experiment Center™ s doprovodným systémem SMI iView X™

a SMI BeGaze™. K nahrávání video a audio záznamů respondentů byla použita webová kamera Logitech.

Vyhodnocení eye-tracking testování a výpočet vizuální salience probíhal v open source programu OGAMA 4.4 (OpenGazeAndMouseAnalyzer). Program vytvořil Adrian Voßkübler z Institut für Didaktik der Physik, Freie Universität Berlin. Pro vyhodnocení výsledků eye-tracking testování interaktivní práce byl použit multimediální přehrávač SMPlayer. Přehledové tabulky byly vytvořeny v open source programu LibreOffice Calc.

Textová část diplomové práce byla napsána v open source programu LibreOffice Writer. Diagramy v textové části práce byly vytvořeny pomocí online aplikace Gliffy. Poster o diplomové práci byl vytvořen v open source programu LibreOffice Impress. K tvorbě webových stránek o diplomové práci byl použit textový editor Geany.

2.3 Postup zpracování

Nejdříve byla nastudována zejména zahraniční literatura a informační zdroje zabývající se vizuální salencí. Poté byla vytvořena rešerše dané problematiky. Dále proběhlo seznámení s vybavením eye-tracking laboratoře na Katedře geoinformatiky a metodou eye-tracking. Přínosný byl návod k programovému vybavení laboratoře *Manuál k produktu SMI Experiment Suite*, který je přílohou bakalářské práce *Implementace nástrojů SMI Experiment Suite 360° pro hodnocení mapové kompozice* (Kotyz, 2013).

Pro praktickou část této práce byly zvoleny čtyři VPL komponenty pro hodnocení vizuální salience. Po podrobném prostudování jednotlivých VPL komponent byly vytvářeny diagramy, které byly následně využity při přípravě eye-tracking testů. Před samotným testováním byly v zimním semestru akademického roku 2013/2014 vyučovány VPL komponenty Model Maker a Macro Modeler.

Testování vizuální salience metodou eye-tracking probíhalo během zimní a letního semestru akademického roku 2013/2014. Celkem bylo vytvořeno pět experimentů.

Pro testování interaktivní práce metodou eye-tracking byla vybrána pouze jedna VPL komponenta ModelBuilder, se kterou mají studenti na Katedře geoinformatiky nejvíce zkušeností. Připraveny byly celkem dva interaktivní testy. První test probíhal v prosinci roku 2013 a dotestován byl v lednu roku 2014. Druhé testování proběhlo v letním semestru akademického roku 2013/2014 během měsíce března.

Praktickou částí této práce bylo také srovnání výsledků testování vizuální salience metodou eye-tracking a vypočítané vizuální salience v programu OGAMA. Proběhlo seznámení s open source programem OGAMA a testování Saliency modulu. Následně byla vypočítána vizuální salience pro vytvořené diagramy všech VPL komponent a poté porovnána s výsledky z eye-tracking testování.

3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Technologie eye-tracking je založena na principu sledování pohybu očí. Jedná se o relativně objektivní metodu, která je často používána pro studie použitelnosti. Mezi významné zahraniční publikace zabývající se metodou eye-tracking patří *Eye tracking: a comprehensive guide to methods and measures* (Holmqvist a kol., 2011) a *Eye tracking methodology: theory and practice* (Duchowski, 2007). Problematiku technologie eye-tracking přehledně shrnuje článek *Eye-tracking a jeho využití při hodnocení map* (Popelka a kol., 2012).

Vizuálním programováním na Katedře geoinformatiky Univerzity Palackého v Olomouci se zabývá v odborných článcích Z. Dobešová. Jedním z nejnovějších příspěvků je *Differences in visual programming for GIS* (Dobešová a Dobeš, 2014). Problematikou VPL komponent se věnuje bakalářská práce *Srovnání vizuálního programování v GIS produktech podle kognitivních dimenzí* (Kudělka, 2013). V rámci této práce byly metodou eye-tracking testovány dvě VPL komponenty Workflow Desinger a ModelBuilder.

3.1 Salience

Salience je cizí slovo, které nemá běžně používaný český ekvivalent. Lze nahradit pojmem výraznost či míra důležitosti (Kohoutek, 2008) nebo také výrazem míra nápadnosti (Wikipedie, 2013). Slovo salience vychází z anglického slova *salient* (význačný, čnicí, vystupující, vyčnívající). V zahraniční literatuře a informačních zdrojích je pro pojem *salience* (Itti, 2007) také často používán výraz *saliency* (Borji, 2013).

Vizuální salience

Vizuální salience (Itti, 2007) lze jednoduše popsat jako výrazné subjektivní vnímání kvality, kdy objekty odlišující se od svého okolí upoutávají naši pozornost. Ale samotná podstata vizuální salience je složitější. Vizuálně nápadné podněty přitahují naši pozornost, nicméně pro mozek je velmi složité zpracovat všechny vizuální vjemy. Tato neschopnost zpracovat najednou všechny atraktivní oblasti lze řešit právě vizuální saliencí. Pokud ale máme zpracovat pouze jeden podnět vyvstane problém, jak ho vybrat. V tomto případě vizuální salience pomáhá mozku provést dostatečně efektivní výběr. Rané stádium vizuálního zpracování podnětu vede k výraznému subjektivnímu vnímání kvality, které umožňuje některým podnětům vyniknout vzhledem k ostatním objektům. Mозek je vyvinutý k tomu, aby rychle a automaticky počítal salience v reálném čase a celém zorném poli. Naše vizuální pozornost je poté přitahována k nápadným místům.

Základem vizuální salience je bezděčný podnětem řízený signál, který oznamuje: „toto místo je dostatečně odlišné od svého okolí a zaslouží si pozornost“. Tento

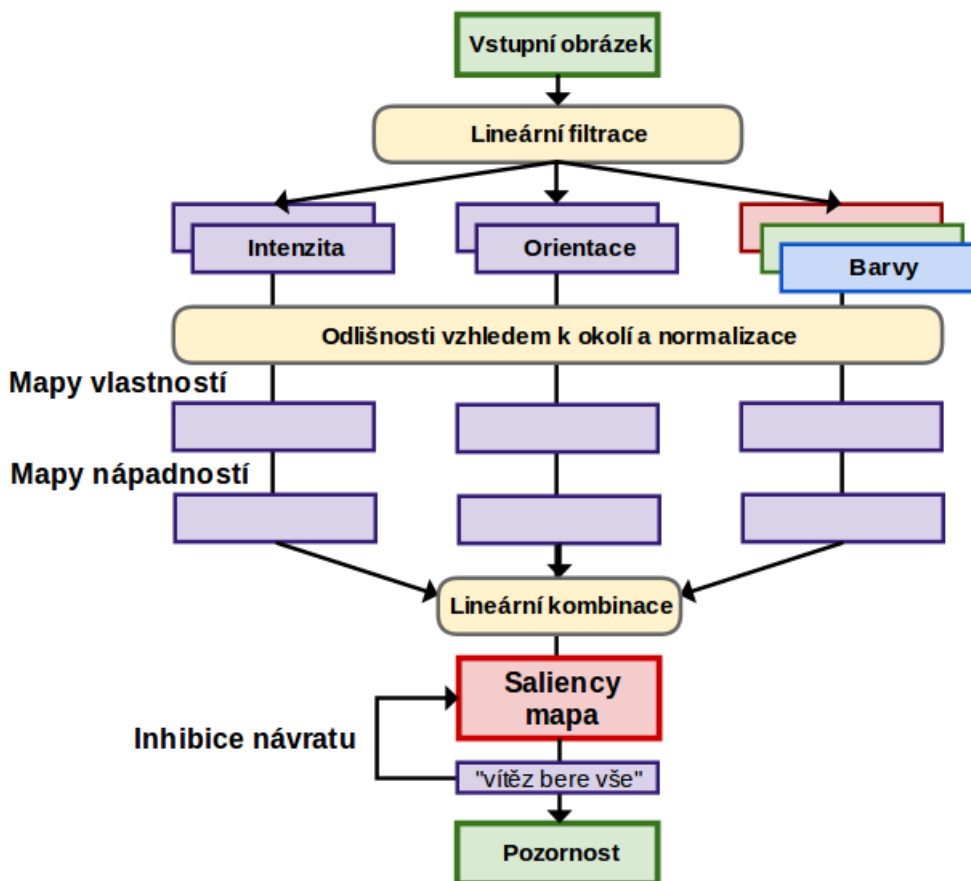
podvědomý signál může být silně modulován nebo dokonce někdy potlačen záměrnými uživatelsky řízenými faktory. Salience tedy závisí jak na předchozích zkušenostech, tak na nevědomí.

Vizuální salience je někdy nesprávně popisována jako fyzická vlastnost vizuálního podnětu. Je důležité vědět, že salience je důsledkem vzájemného působení podnětů.

Saliency mapa

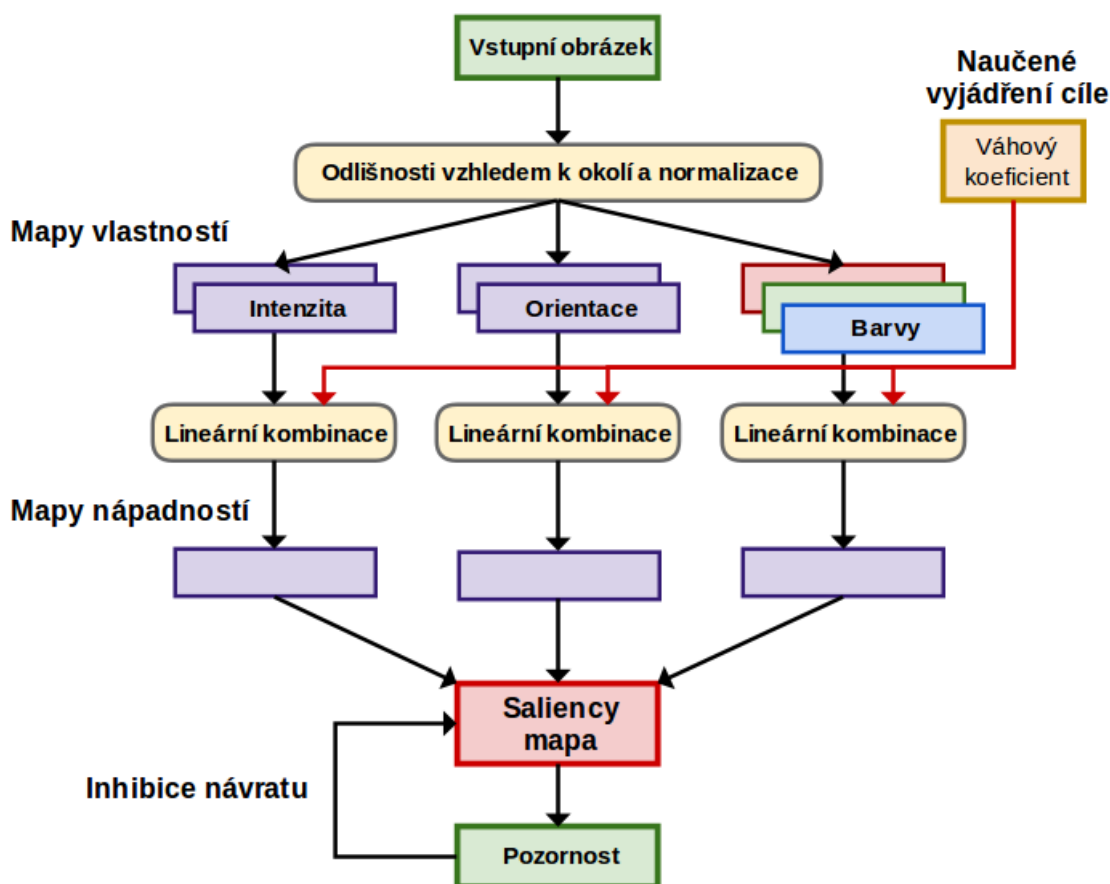
Niebur (2007) definuje saliency mapu jako topograficky uspořádanou mapu, která představuje vizuální salience odpovídající vizuálnímu vjemu. Jedná se o lineární kombinaci jednotlivých vizuálních vlastností (mapy vlastností tzv. feature maps), které přispívají k výběru nápadného podnětu, jako je například barva, orientace, intenzita nebo pohyb. Existují dva základní druhy saliency modelů, podle kterých lze vypočítat výslednou saliency mapu (viz Příloha 3).

- Bezděčný saliency model tzv. bottom-up saliency model (Obr. 3.1)
- Záměrný saliency model tzv. top-down saliency model (Obr. 3.2)



Obr. 3.1 Bezděčný saliency model tzv. bottom-up saliency model (upraveno podle Itti a kol., 1998).

Srovnáním saliency modelů se podrobně věnuje Borji a kol. (2013). V současné době existuje 35 dostupných modelů. První saliency model byl uveřejněn v roce 1998 a jednalo se o bezděčný saliency model (Itti a kol., 1998).



Obr. 3.2 Záměrný saliency model tzv. top-down saliency model (upraveno podle Navalpakkam a Itti, 2005).

Výpočet salience

- ILab Neuromorphic Vision C++ Toolkit (iNVT) – <http://ilab.usc.edu/toolkit/>
- Saliency Module v programu OGAMA – <http://www.ogama.net/> (viz kap. 5.1)
- SaliencyToolbox – <http://www.saliencytoolbox.net/>

ILab Neuromorphic Vision C++ Toolkit je sada C++ tříd pro vývoj neuromorfních modelů vizuální pozornosti. Jedná se o výpočetní algoritmy neurovědy, jejichž architektura a funkce vycházejí z poznatků o lidských mozcích. Mimo jiné iNVT zahrnuje plně vyvinutý bezděčný saliency model (tzv. bottom-up saliency model), který je implementován i do Saliency modulu v programu OGAMA (viz kap. 5.1). Zdrojové kódy jsou volně šiřitelné pod licencí GNU General Public License.

SaliencyToolbox je sada MATLAB funkcí a skriptů pro výpočet saliency map obrazů. Jedná se o částečnou reimplementaci iNVT toolkitu, zejména jde o redukci jeho zdrojového kódu (Walther, 2013). Toolbox je implementován do programu

MATLAB[®], což se projevuje na rychlosti výpočtu salience, která je pomalejší než u iNVT. K využití SaliencyToolboxu je potřeba splnit následující systémové požadavky, mezi které patří zejména nainstalované produkty MATLAB[®] a Image Processing Toolbox[™] od americké společnosti MathWorks. Výhodou toolboxu je jeho využití v různých operačních systémech, které jsou podporovány programem MATLAB[®].

Salience a vizuální programování

Jednou ze základních vlastností programovacích jazyků je srozumitelnost, kterou může ovlivňovat řada faktorů (Marriott a Meyer, 1998). Díky použití grafických prvků jsou vizuální programovací jazyky lépe pochopitelné než textové programovací jazyky. Jedním z faktorů, který může ovlivňovat srozumitelnost vizuálních programovacích jazyků je salience. Ta je mírou významně nápadné podoby grafických prvků.

Například, když se používá jako grafický prvek kruh, může mít několik vizuálních parametrů: umístění, poloměr, tloušťka obrysu, barva obrysu, barva výplně apod. Z n možných parametrů v D , jestli jazyk využívá m parametrů, z kterých pouze l má smysl, pak míra nápadnosti D pro kruhy lze vyjádřit poměrem l/m . Nízká salience ukazuje na vysoký počet nepodstatných a potenciálně rozptylujících vizuálních parametrů podoby grafických prvků, což vede k malé srozumitelnosti. U vizuálních programovacích jazyků s nízkou mírou nápadnosti by se proto mělo uvažovat o úpravě jednotlivých grafických prvků, tak aby bylo dosaženo lepší srozumitelnosti.

Snadnost čtení diagramů datových toků také nepříznivě ovlivňuje vysoký počet křížení spojovacích šipek, které propojují jednotlivé grafické prvky.

4 TESTOVÁNÍ SALIENCE METODOU EYE-TRACKING

Pro hodnocení vizuální salience byly vybrány čtyři VPL komponenty: Model Maker, Macro Modeler, ModelBuilder a Workflow Designer. Tyto VPL komponenty byly zvoleny s ohledem na to, že jsou součástí programů používaných na Katedře geoinformatiky Univerzity Palackého v Olomouci.

VPL komponenty ModelBuilder a Workflow Designer jsou na Katedře geoinformatiky běžně vyučovány na rozdíl od vybraných VPL komponent Model Maker a Macro Modeler. Z tohoto důvodu probíhala v zimním semestru akademického roku 2013/2014 výuka VPL komponent Model Maker a Macro Modeler v rámci předmětu KGI/DAMIN Data mining. Účastníci tohoto kurzu byli studenti 1. ročníku navazujícího magisterského oboru Geoinformatika studující na Katedře geoinformatiky Univerzity Palackého v Olomouci. Příprava praktických cvičení pro studenty a následná výuka probíhala ve spolupráci s kolegyní Janou Zieglerovou.

V říjnu roku 2013 proběhlo testování VPL komponenty Model Maker na vizuální salience pomocí technologie eye-tracking. Celkem byla zaznamenána data od 18 respondentů. Ale jeden respondent (P11) byl kvůli velkým odchylkám při kalibraci vyřazen. Z tohoto důvodu byla analyzována data od 17 respondentů. V listopadu téhož roku bylo uskutečněno testování VPL komponenty Macro Modeler, kterého se zúčastnilo 19 respondentů. V prosinci roku 2013 následovalo eye-tracking testování VPL komponenty ModelBuilder, které bylo následně dotestováno s ohledem na časové možnosti studentů v měsíci březnu roku 2014. Testování VPL komponenty ModelBuilder se kromě studentů 1. ročníku navazujícího magisterského oboru Geoinformatika zúčastnili také studenti 2. ročníku bakalářského oboru Geoinformatika a geografie v rámci předmětu KGI/SKRI Skriptování v GIS. Celkem se testování VPL komponenty zúčastnilo 21 respondentů. Poslední VPL komponenta Workflow Designer byla otestována v měsíci dubnu roku 2014 v rámci předmětu KGI/CADX CAD, který navštěvovali studenti 1. ročníku navazujícího magisterského oboru Geoinformatika. Testování této VPL komponenty se zúčastnili také studenti 2. ročníku navazujícího magisterského oboru Geoinformatika, který navštěvovali kurz v akademickém roce 2012/2013. Celkem bylo otestováno 19 respondentů.

Během testování jednotlivých VPL komponent bylo rozhodnuto zahrnout do testování také respondenty bez zkušeností s vizuálními programovacími jazyky. Pro tyto respondenty byl vytvořen nový experiment, který obsahoval diagramy všech čtyř VPL komponent z předchozích testování. Osloveni byli zejména studenti 1. ročníku bakalářského oboru Geoinformatika a geografie studující na Katedře geoinformatiky. Testování proběhlo v měsíci březnu roku 2014 a celkem se ho zúčastnilo 21 respondentů. Výsledky byly následně porovnány s výsledky respondentů, kteří měli zkušenosti s VPL komponentami.

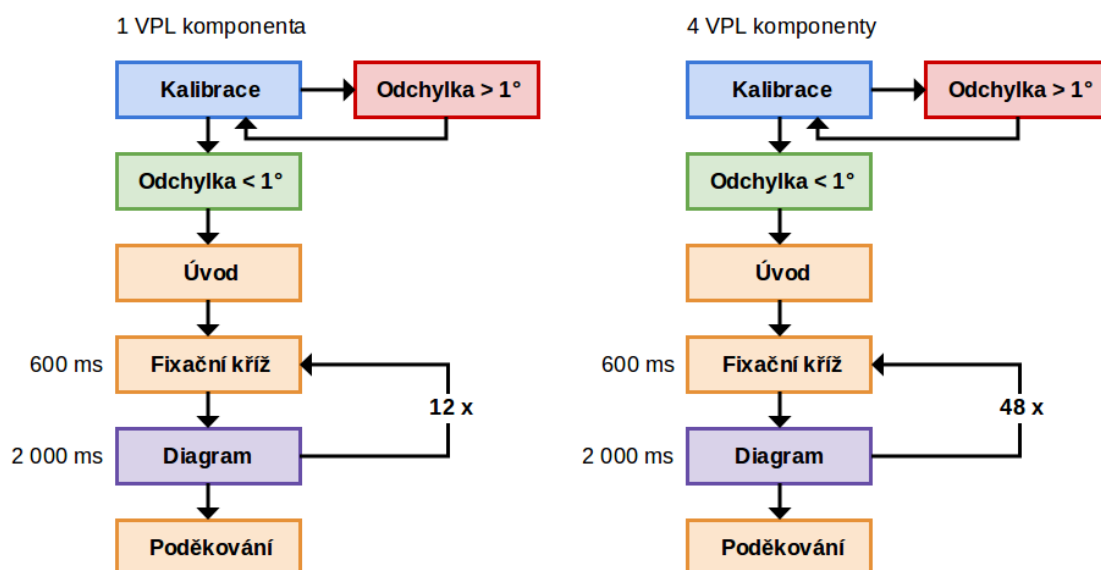
Tab. 4.1 Počty respondentů testování vizuální salience metodou eye-tracking ve vybraných VPL komponentách

VPL komponenta	Program	Počet respondentů	
		Zkušenosti s VPL	Bez zkušeností s VPL
Model Maker	ERDAS Imagine 2013	18	21
Macro Modeler	Idrisi 17.0 The Selva Edition	19	
ModelBuilder	ArcGIS 10.1 for Desktop Advanced	21	
Workflow Designer	Autodesk AutoCAD Map 3D 2014	19	

4.1 Struktura experimentů

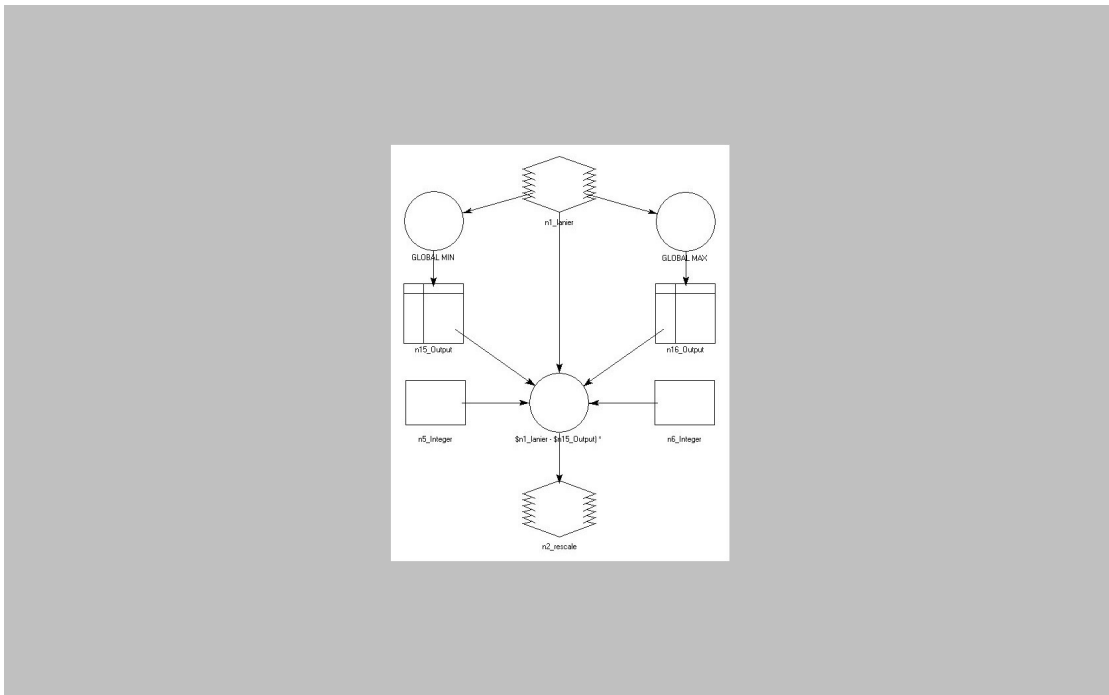
Pro každou VPL komponentu byl vytvořen samostatný experiment. Testování se účastnili studenti se zkušenostmi s těmito VPL komponentami. Pro respondenty bez zkušeností s VPL byl vytvořen další experiment, který obsahoval diagramy všech čtyř VPL komponent z předchozích testování (viz Obr. 4.1). Celkem bylo tedy vytvořeno pět experimentů pro testování salience.

Struktura experimentů pro jednotlivé VPL komponenty byla obdobná. Každý test obsahoval 12 experimentálních stimulů v podobě obrázků, které se měnily po dvou sekundách. Před každým diagramem byl zobrazen fixační kříž po dobu 600 ms (viz Obr. 4.1), aby byla zaručena pro všechny respondenty shodná výchozí pozice při sledování stimulů. Experiment, obsahující diagramy všech čtyř VPL komponent, se lišil pouze počtem experimentálních stimulů, kterých bylo celkem 48.



Obr. 4.1 Struktura experimentů pro testování vizuální salience metodou eye-tracking.

Pro testování VPL komponenty Model Maker byl použit monitor s rozlišením obrazovky 1680 x 1050 bodů, u dalších testování byl pak využit displej s rozlišením obrazovky 1920 x 1200 bodů. Pozice očí byly snímány vzorkovací frekvencí 120 Hz. V prvních dvou testech (Model Maker a Macro Modeler) tvořily experimentální stimuly diagramy s výchozím pozadím používaným ve VPL komponentách. Rozlišení obrázků s diagramy, ale zdaleka nedosahovalo rozlišení obrazovky, proto byl zbytek plochy obrazovky vyplněn výchozím šedým pozadím, které nelze v SMI Experiment Center™ přenastavit (viz Obr. 4.2). Po uvážení, že přechod mezi pozadím diagramu (zpravidla bílé) a šedým pozadím může respondenta rušit, byl pro testování dalších dvou VPL komponent (ModelBuilder a Workflow Designer) zvolen jiný přístup. Obrázky jednotlivých diagramů byly pomocí grafického editoru upraveny na jednotné rozlišení, které bylo shodné s rozlišením obrazovky. Pozadí výsledného obrázku bylo vyplněno barvou pozadí diagramu. Všechny diagramy jsou přiloženy na DVD, jak ve formě obrázků, tak souborů použitelných v daných VPL komponentách.



Obr. 4.2 Ukázka experimentálního stimulu pro testování vizuální salience VPL komponenty Model Maker metodou eye-tracking.

5 HODNOCENÍ VIZUÁLNÍ SALIENCE

Jedním z hlavních cílů práce bylo eye-tracking vyhodnocení salience ve vizuálních programovacích jazycích používaných v GIS. Po testování vizuální salience metodou eye-tracking přišlo na řadu srovnání výsledků eye-tracking testování a vypočítané vizuální salience pomocí programu OGAMA. Nejdříve bylo nutné provést seznámení s open source programem OGAMA a otestovat Saliency modul. Následně byla v tomto programu vypočítána vizuální salience pro všechny vytvořené diagramy datových toků ve čtyřech vybraných VPL komponentách. Poté byla generovaná salience porovnána s naměřenými daty z eye-tracking testování salience. Na závěr byly srovnány výsledky, eye-tracking testování vizuální salience, respondentů se zkušenostmi s VPL komponentami a bez zkušeností s VPL komponentami.

5.1 OGAMA (Saliency Module)

OpenGazeAndMouseAnalyzer je open source program určený pro přípravu, průběh a vyhodnocování eye-tracking testů. Aplikace obsahuje několik modulů. Cílem práce bylo podrobněji otestovat zejména Saliency modul. K programu je dostupný pouze strohý návod pro verzi 2.5 (Voßkübler, 2009). Aktuální verze programu je 4.4.

Saliency modul je určen pro výpočet míry nápadnosti na experimentálních stimulech. Saliency modul využívá iLab Neuromorphic Vision C++ Toolkit (iNVT), který vyvíjí Laurent Itti na americké výzkumné univerzitě University of Southern California v Los Angeles. Je založen na saliency modelu, který je postaven na bezděčné pozornosti tzv. bottom-up saliency model (Obr. 3.1).

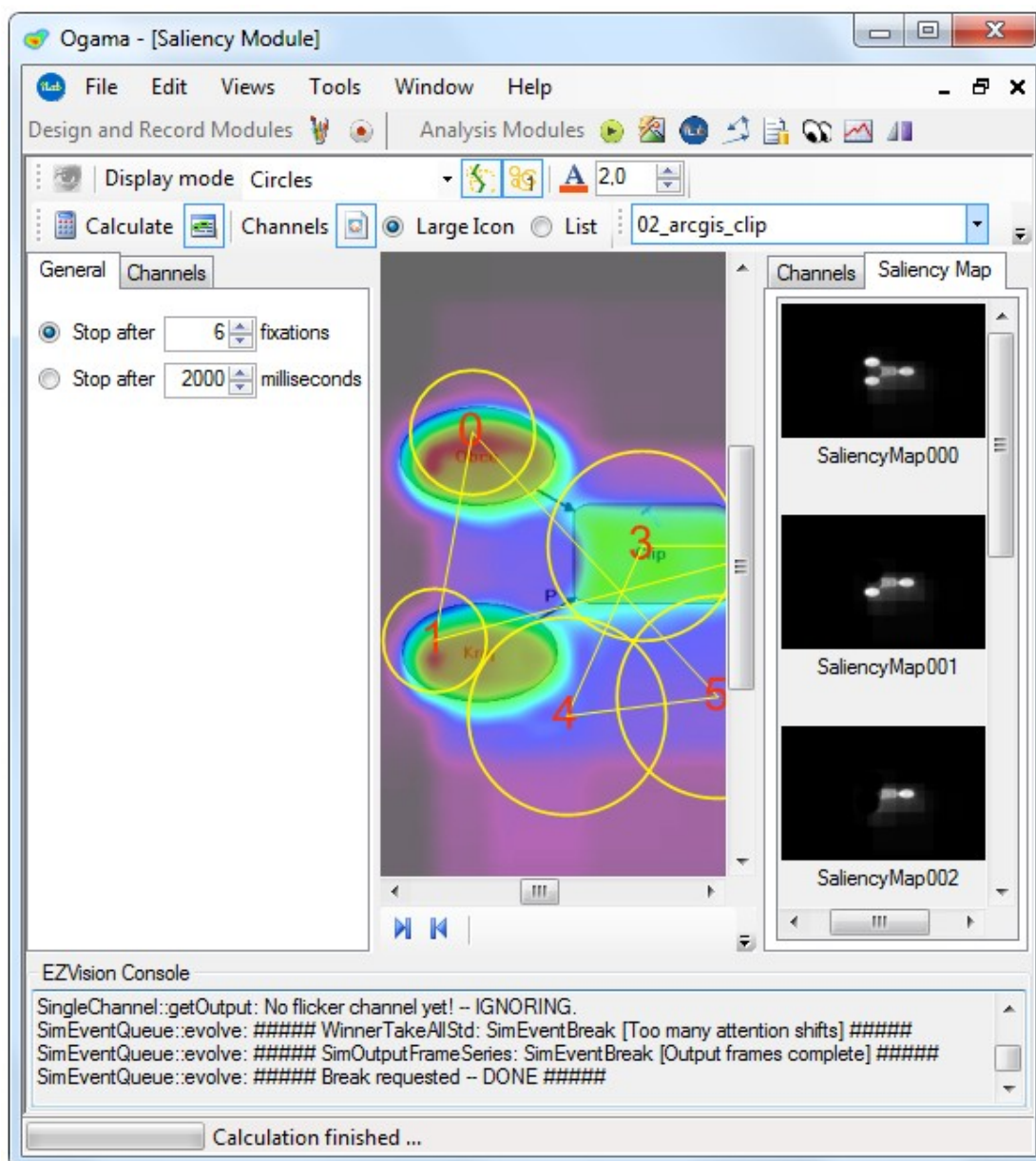
Metoda Saliency je schopna predikovat scanpath (trajektorie pohledu) a vypočítat saliency mapy. Snahou bylo zjistit jaké konkrétní algoritmy jsou v programu pro výpočet salience používány. Saliency modul v nápovědě odkazuje na stránky autora tohoto modulu <http://ilab.usc.edu/toolkit/>, kde jsou k dispozici přímo zdrojové kódy. Nelze ale zjistit, které části byly do programu OGAMA implementovány a co přesně znamenají jednotlivá nastavení, zejména pak parametry kombinace kanálů. Proto byla metoda Saliency otestována na vybraných stimulech.

Před samotným spuštěním výpočtu míst s největší mírou nápadnosti je potřeba provést nastavení jednotlivých parametrů. Základní nastavení se provádí v záložce *General*, kde lze nastavit, jak má být výpočet zastaven. Lze vybírat ze dvou možností (viz Obr. 5.1):

- zastavení po zadaném počtu fixací,
- zastavení po zadaném časovém údaji v milisekundách.

Při testování modulu bylo zjištěno, že výsledkem je vždy shodná saliency mapa bez ohledu na to, jak má být výpočet zastaven. Pokud ovšem zůstane zachováno nastavení parametrů v záložce *Channels*. Rozdíl je pouze v rostoucím počtu

predikovaných fixací, kdy počáteční a následující fixace zůstávají pozicí a velikostí beze změny. Pro více fixací jsou generovány jen další nové fixace na dosud neobsazených místech.

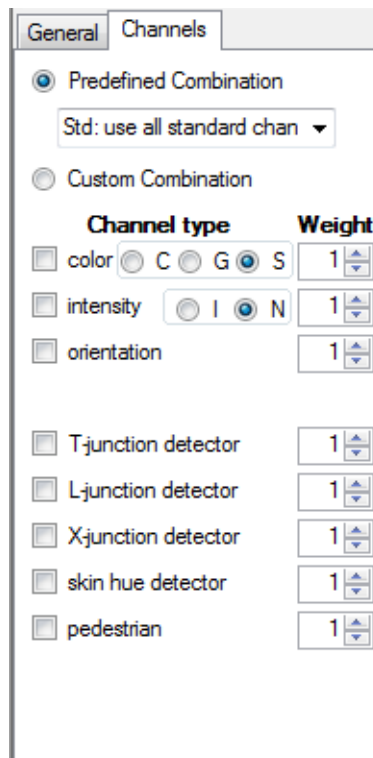


Obr. 5.1 Prostředí programu OGAMA – Saliency modul.

Výsledná saliency mapa je kombinací jednotlivých kanálů, které lze nastavit v záložce *Channels* (viz Obr. 5.2). Lze využít buď předdefinované kombinace kanálů (*Predefined Combination*) nebo vytvořit vlastní kombinace jednotlivých kanálů (*Custom Combination*). Předdefinované kombinace kanálů nabízí na výběr z pěti možností (viz Tab. 5.1 a Příloha 3).

Nastavení vlastní kombinace kanálů nabízí na výběr z osmi typů kanálů (*Channel type*), kterým lze přiřadit různé váhy (*Weight*). Třemi hlavními kanály jsou barva

(Color), intenzita (*Intensity*) a orientace (*Orientation*). U kanálu barva, lze dále zvolit tři druhy: C, G, S a u kanálu intenzita dva: I a N (viz Tab. 5.2 a Příloha 3).



Obr. 5.2 Záložka *Channels* pro nastavení parametrů kombinace kanálů v programu OGAMA – Saliency modul.

Tab. 5.1 Předdefinované kombinace kanálů a jejich výstupy v programu OGAMA – Saliency modul

Název kombinace kanálů	Výstupy	
	Kanál (typ)	Saliency mapa
Std: use all standard channels with unit weights	Color (C), Orientation, Intensity (I)	ano
SurpStd: use all standard Surprise channels	Color (C), Orientation, Intensity (I)	ano
PN03contrast: Parkhurst & Niebur'03 contrast model	ne	ano
Variance: local variance in 16x16 image patches	ne	ano
Scorr: spatial correlation between image patches in a frame	ne	ano

Tab. 5.2 Vlastní kombinace kanálů a jejich výstupy v programu OGAMA – Saliency modul

Název kanálů (typ)	Výstupy	
	Kanál (typ)	Saliency mapa
Color (S)	Color (C), Color (S1), Color (S2)	ano (shodná s Color (S1))
Color (G)	Color (G)	ano (shodná s Color (G))
Color (C)	Color (C)	ano (shodná s Color (C))
Intensity (N)	Intensity (N)	ano (shodná s Intensity (N))
Intensity (I)	ne	ano
Orientation	Orientation	ano (shodná s Orientation)
T-junction detector	Orientation, T-junction	ano (shodná s T-junction)
L-junction detector	Orientation, L-junction	ano (shodná s L-junction)
X-junction detector	Orientation, X-junction	ano (shodná s X-junction)

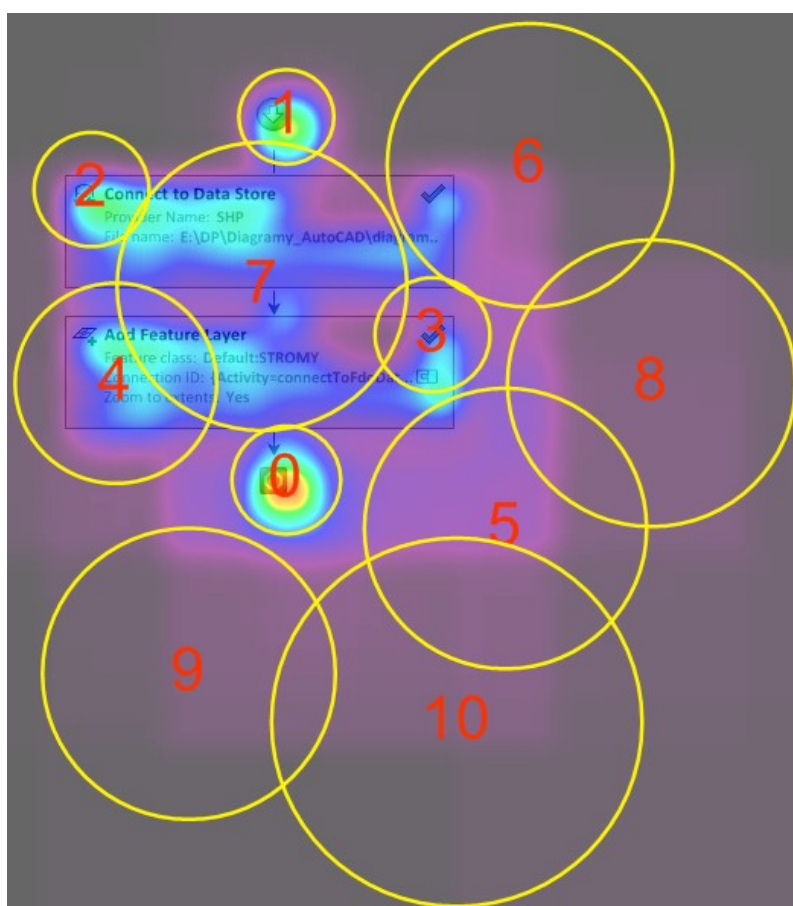
5.2 Výpočet salience pro diagramy datových toků

Dalším krokem byl výpočet vizuální salience pro všechny experimentální stimuly z testování salience metodou eye-tracking (diagramy čtyř vybraných VPL komponent). Vizuální salience pro jednotlivé diagramy datových toků byla generována v programu OGAMA pomocí Saliency modulu. V tomto programu byla také analyzována data naměřená testováním salience metodou eye-tracking. Pro obě činnosti byly použity shodné experimentální stimuly, proto mohly být použity stejné experimenty programu OGAMA. Z toho důvodu bylo potřeba exportovat naměřená data ze všech pěti testování salience metodou eye-tracking pomocí produktu BeGaze™. Následně byly pro jednotlivá testování vytvořené experimenty v programu OGAMA, kde byla naměřená data importována. Všem pěti experimentům byly nastaveny shodné parametry (viz Obr. 5.5), pouze u experimentu pro VPL komponentu Model Maker se lišilo rozlišení obrazovky (1680 x 1050 bodů).

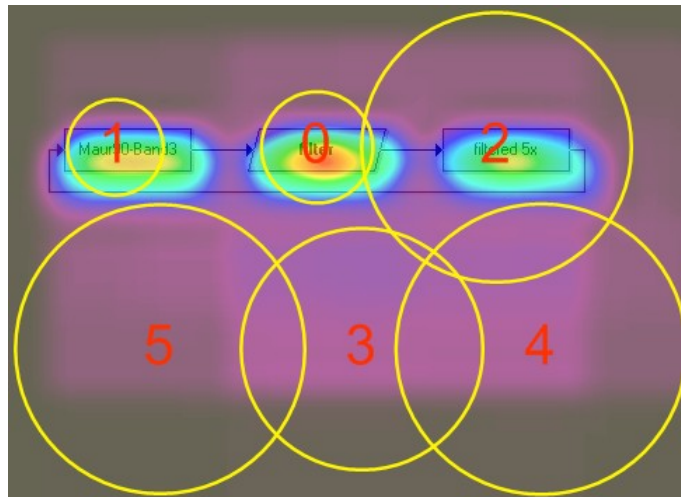
Při testování Saliency modulu bylo cílem zjistit, které kombinace kanálů budou vyhovující pro následné srovnání vypočítané salience s naměřenými daty metodou eye-tracking. Jako nejvhodnější možnost byla vybrána předdefinovaná kombinace kanálů *Std: use all standard channels with unit weights*. Toto nastavení bylo použito pro výpočet vizuální salience pro všechny experimentální stimuly všech testovaných VPL komponent.

Dále bylo potřeba vybrat způsob zastavení výpočtu salience. Nejdříve bylo vybráno zastavení po zadaném časovém údaji. Snahou bylo zadat shodný časový údaj jako při sledování stimulů u eye-tracking testování, tedy nastavit 2 000 ms. Tato

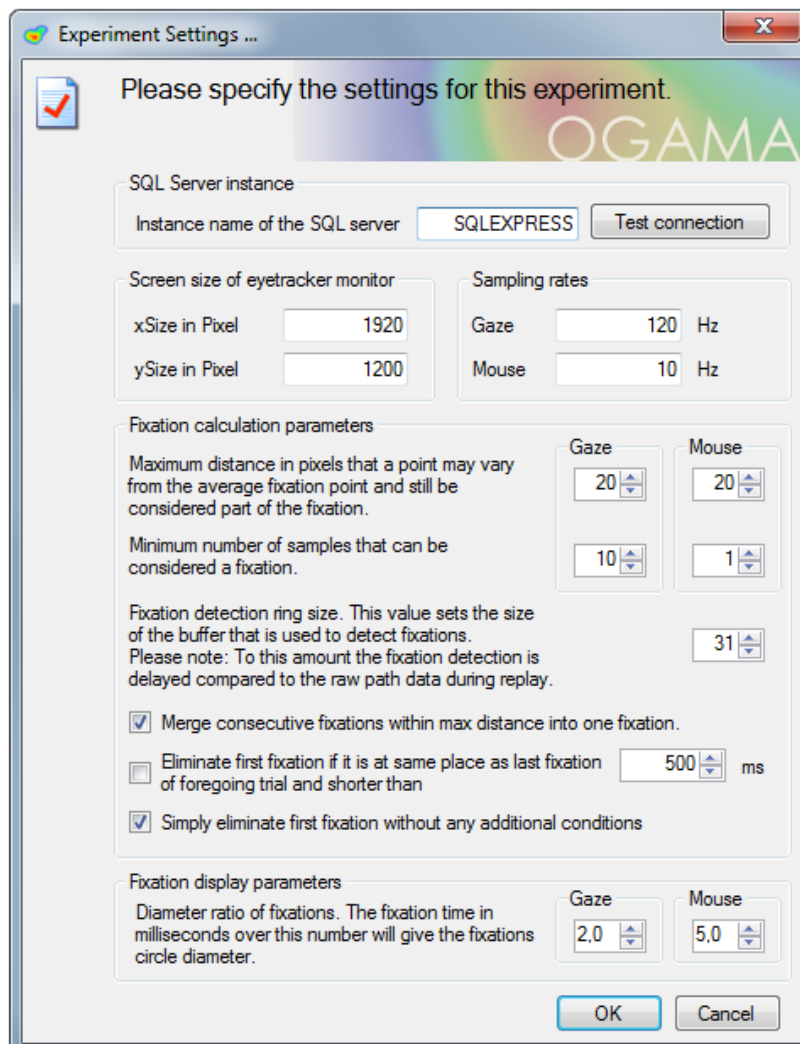
varianta nebyla vhodná, výsledkem byl příliš velký počet fixací (viz Obr. 5.3). Ty se nacházely i mimo diagram. Proto byla zvolena druhá možnost nastavení: zastavení po zadaném počtu fixací. Pro výpočet počtu fixací byl odvozen následující postup. Nejdříve byly zjištěny počty fixací respondentů u jednotlivých experimentálních stimulů z naměřených dat metodou eye-tracking. Poté byl vypočítán pro jednotlivé stimuly medián. Aby byla zohledněna první fixace, která je díky fixačnímu kříži zpravidla uprostřed stimulu, byla od mediánu odečtena hodnota jedna. Výsledné hodnoty byly následně použity pro nastavení parametru zastavení po zadání počtu fixací, ten se u jednotlivých experimentálních stimulů lišil. S tímto nastavením byla vypočítána salience pro všechny experimentální stimuly všech testovaných VPL komponent. Počet zadaných fixací byl i přesto ve většině případů stále příliš velký (viz Obr. 5.4), proto bylo potřeba zvolit menší hodnotu s ohledem na potřeby jednotlivých diagramů. Zejména u diagramů s nízkým počtem prvků neměl vysoký počet fixací smysl. Výsledky vypočítané vizuální salience byly vizualizovány ve formě saliency map s predikovaným pořadím fixací. Všechny výsledky jsou součástí příloh diplomové práce (Příloha 6).



Obr. 5.3 Saliency mapa s pořadím fixací vypočítána s nastavením předdefinované kombinace kanálů *Std*, výpočet zastaven po 2 000 ms (Workflow Designer).



Obr. 5.4 Saliency mapa s pořadím fixací vypočítána s nastavením předdefinované kombinace kanálů *Std*, výpočet zastaven po šesti fixacích (Macro Modeler).



Obr. 5.5 Nastavení parametrů experimentů v programu OGAMA.

5.3 Srovnání výsledků eye-tracking testování a vypočítané salience

Jedním z cílů diplomové práce bylo porovnání vypočítané vizuální salience v programu OGAMA a výsledků testování salience metodou eye-tracking. S ohledem na rozsah práce a velké množství diagramů testovaných metodou eye-tracking (celkem 48 experimentálních stimulů) byly pro následující textovou část vybrány dva diagramy pro každou VPL komponentu (celkem osm experimentálních stimulů). Jednotlivé výsledky srovnání byly vedle vizualizace také slovně okomentovány.

Data naměřená pomocí technologie eye-tracking byla vizualizována v podobě heatmap (fixační mapa). Cílem bylo zjistit, co upoutá pozornost respondentů na první pohled. Z tohoto důvodu byla snaha eliminovat první fixaci, která byla vždy uprostřed stimulu, díky fixačnímu kříži, který byl zařazen mezi jednotlivé stimuly pro zachování shodné výchozí pozice pro jednotlivé respondenty. V nastavení jednotlivých experimentů byla zvolena možnost eliminovat první fixaci (viz Obr. 5.5), bohužel ale bylo zjištěno, že v heatmapě se toto nastavení nijak neprojeví. Proto bylo při srovnání salience sledována především druhá, případně třetí fixace.

Vizuální salience pro jednotlivé experimentální stimuly byla vypočítána s nastavením předdefinované kombinace kanálů *Std: use all standard channels with unit weights* a nejvhodnějším počtem fixací (viz kap. 5.2). Vypočítaná salience byla zobrazena ve formě saliency mapy zároveň s pořadím predikovaných fixací (scanpath bez sakád). Pořadí předpokládaných fixací je počítáno vždy od nuly.

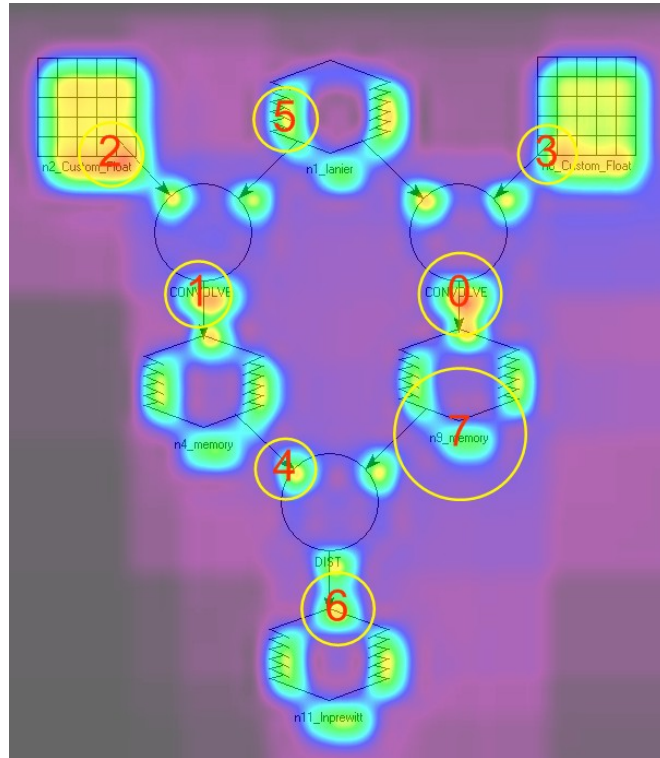
Důležité je také zmínit, že pomocí Saliency modulu lze vypočítat predikovaný scanpath, ale předpokládané pořadí fixací závisí na tom, kde se již nachází předchozí fixace. Ve většině případů se predikované fixace zobrazí na místě, kde zatím žádná fixace nebyla. Po zadání neúměrného počtu fixací na začátku výpočtu či po vyčerpání vhodných pozic se zobrazí fixace i mimo zájmovou oblast (viz Obr. 5.3 a 5.4). Na rozdíl od respondenta, který se může pohledem i několikrát vrátit na stejné místo. U respondenta je také málo pravděpodobné, že bude sledovat okolí samotného diagramu.

5.3.1 Model Maker

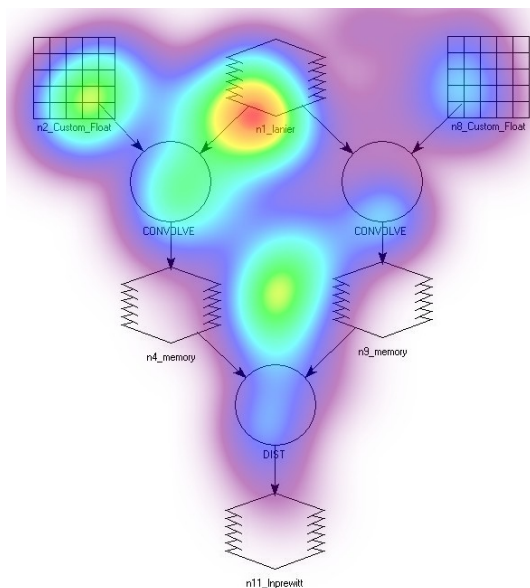
Ve VPL komponentě Model Maker jsou využívány k sestavování diagramů černé objekty na bílém pozadí. Z toho důvodu pro tyto diagramy datových toků nelze vypočítat barevný kanál. Oproti tomu respondenti při testování metodou eye-tracking vnímají i tyto barvy. Výsledná saliency mapa s nastavením *Std: use all standard channels with unit weights* je tedy kombinací pouze dvou kanálů (orientace a intenzita).

U diagramu *05_erdas* (Obr. 5.8) byla největší salience vypočítána u prvků typu matice (fixace 2 a 3) a dále u názvů dvou shodných nástrojů *Convolve* (viz Obr. 5.6). Respondenty, za dobu dvou sekund, kdy sledovali diagram, nejvíce zaujal vstupní

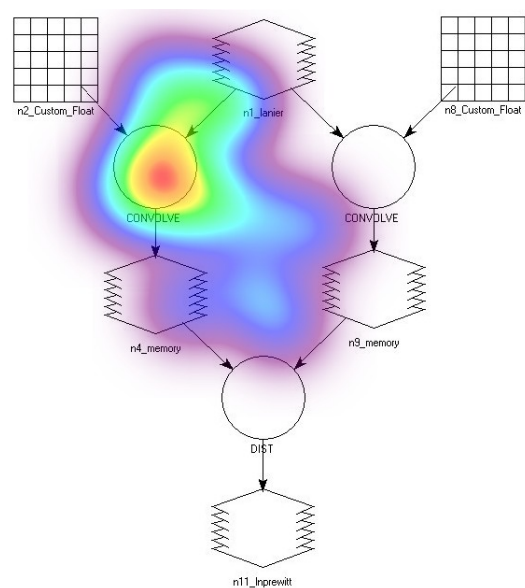
rastr (viz Obr. 5.7). Ale z druhé fixace (viz Obr. 5.8) je zřejmé, že nejdříve jejich pohled připoutal nástroj *Convolve*. Nejpravděpodobněji se, ale respondenti snažili začít prohlížet diagram od vstupních vrstev (začátek diagramu), v tomto případě tedy shora dolů. Toto tvrzení bylo jasně potvrzeno u diagramu *08_erdas* (Obr. 5.10), kdy největší salience byla vypočítána pro výstupní vrstvu (viz Obr. 5.9), ale respondenti nejvíce sledovali vstupní rastr (viz Obr. 5.10).



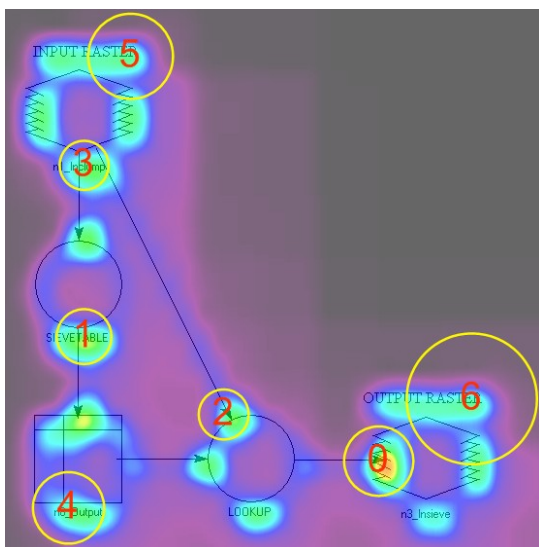
Obr. 5.6 Diagram *05_erdas* – saliency mapa a pořadí osmi fixací (Model Maker).



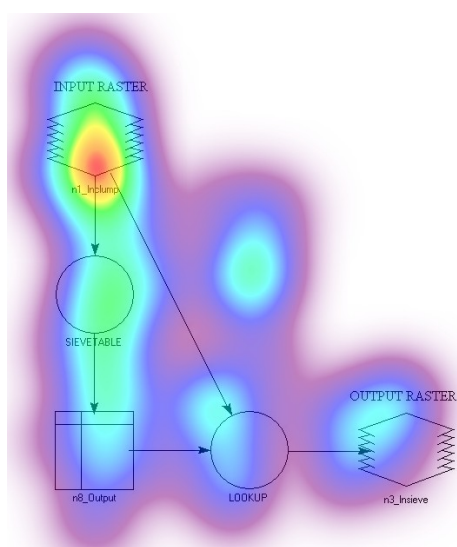
Obr. 5.7 Diagram *05_erdas* – heatmapa všech fixací (Model Maker).



Obr. 5.8 Diagram *05_erdas* – heatmapa druhé fixace (Model Maker).



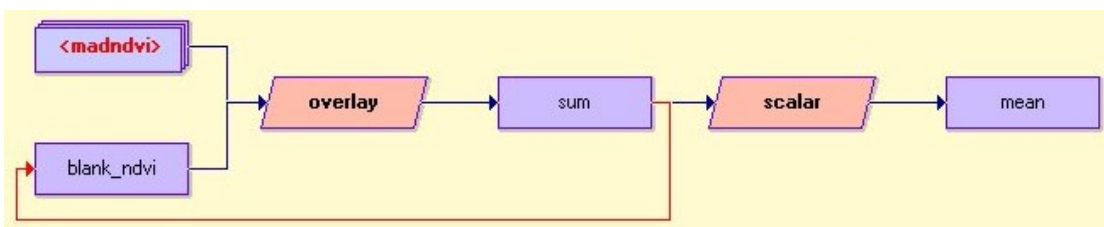
Obr. 5.9 Diagram 08_erdas – saliency mapa a pořadí sedmi fixací (Model Maker).



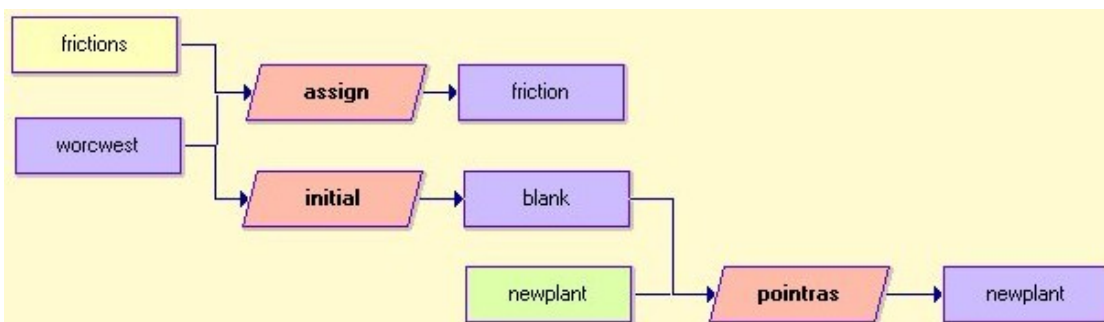
Obr. 5.10 Diagram 08_erdas – heatmapa všech fixací (Model Maker).

5.3.2 Macro Modeler

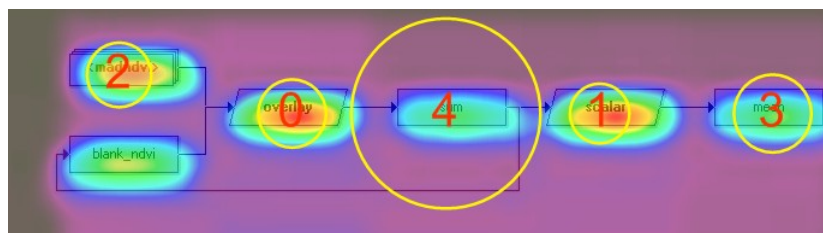
Typické pro prostředí VPL komponenty Macro Modeler je světle žluté pozadí jednotlivých diagramů, u ostatních VPL komponent je pozadí bílé. Jednotlivé prvky diagramu se mezi sebou na první pohled liší pouze barvou (Obr. 5.11 a 5.12). Nástroje jsou zobrazovány v podobě červených kosodélníků, vstupní a výstupní data ve formě obdélníků fialové barvy. Další významnou odlišností jsou popisy jednotlivých prvků.



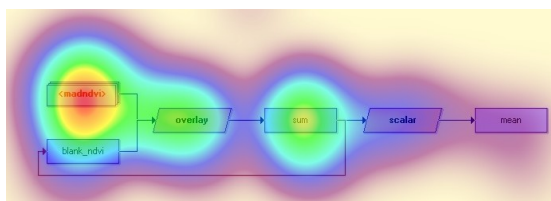
Obr. 5.11 Diagram 06_idrisi (Macro Modeler).



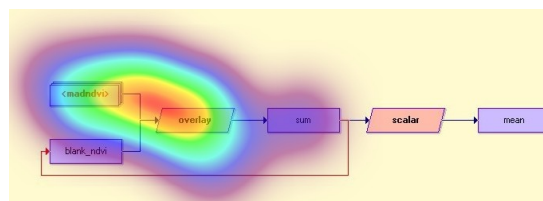
Obr. 5.12 Diagram 12_idrisi (Macro Modeler).



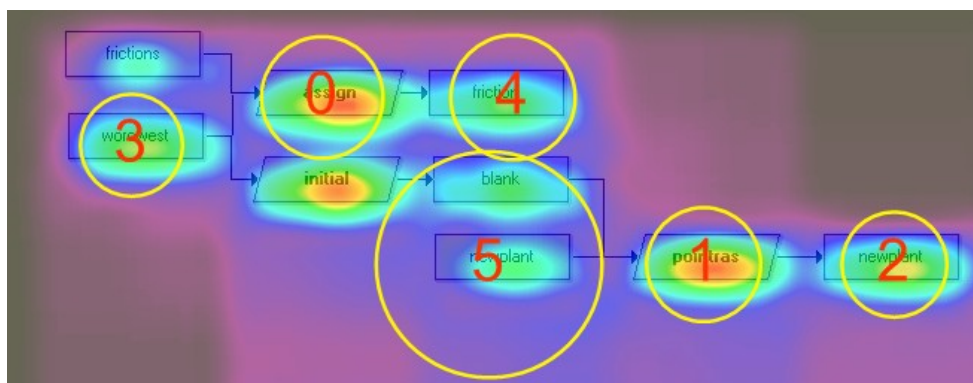
Obr. 5.13 Diagram *06_idrisi* – saliency mapa a pořadí pěti fixací (Macro Modeler).



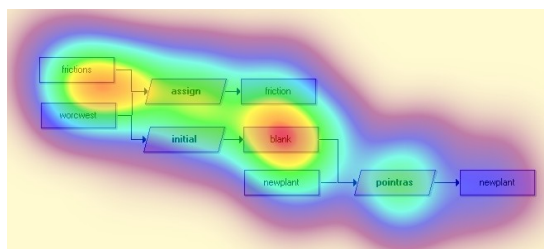
Obr. 5.14 Diagram *06_idrisi* – heatmapa všech fixací (Macro Modeler).



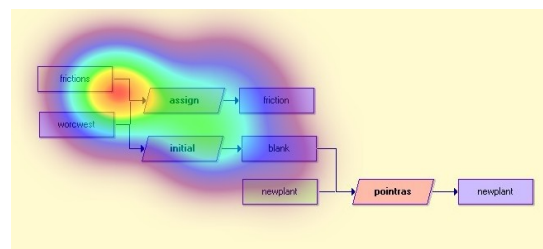
Obr. 5.15 Diagram *06_idrisi* – heatmapa druhé fixace (Macro Modeler).



Obr. 5.16 Diagram *12_idrisi* – saliency mapa a pořadí šesti fixací (Macro Modeler).



Obr. 5.17 Diagram *12_idrisi* – heatmapa všech fixací (Macro Modeler).



Obr. 5.18 Diagram *12_idrisi* – heatmapa druhé fixace (Macro Modeler).

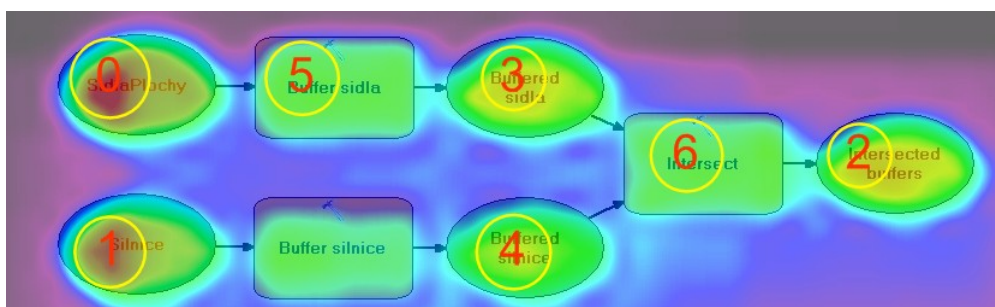
Pro diagram *06_idrisi* (Obr. 5.11) byla vypočítaná největší salience pro nástroje (fixace 0 a 1), dále pak pro dynamickou skupinu (viz Obr. 5.13). Lze tedy tvrdit, že nápadné jsou především ty prvky s popisem, který je proveden tučným písmem. Respondenty nejvíce upoutal prvek typu dynamická skupina (viz Obr. 5.14), což se shoduje s třetí predikovanou fixací. V tomto případě nelze jednoznačně určit, jestli byl prvek nápadnější než ostatní nebo se respondenti pouze snažili začít číst diagram zprava doleva (tedy od vstupních dat k výstupu). Pravděpodobnější je druhá varianta,

kteřou může potvrdit nástroj *scalar*. Pro tento prvek byla predikována druhá fixace, ale respondenty tento objekt vůbec nezaujal (viz Obr. 5.15).

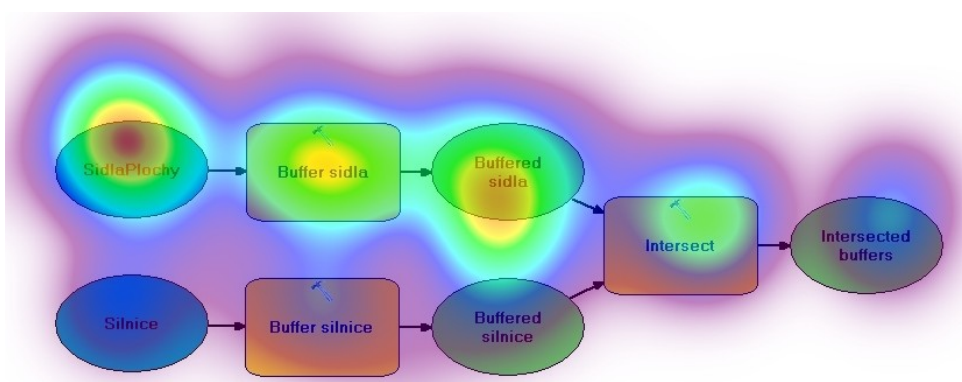
Diagram *12_idrisi* (Obr. 5.12) potvrzuje domněnku, že respondentův pohled je ve velké míře ovlivněn způsobem čtení diagramu neboli směrem čtení diagramu. Vstupní soubor atributových hodnot *frictions* je reprezentován obdélníkem žluté barvy. Stejný barevný tón má i pozadí diagramu. Z tohoto důvodu byla pro tento prvek vypočtena nízká salience a lze ho tak označit za nejméně nápadný prvek diagramu, pokud nejsou brány v úvahu spojovací šipky. Ale respondenti prvek typu soubor atributových hodnot vyhledávali mezi prvními (viz Obr. 5.17), i když není příliš nápadný. Z jejich pohledu se tedy jedná o významný prvek, kterým začíná diagram. Oproti tomu poslední dva prvky v diagramu nástroj *pointtras* a výstup *newplant* respondenty vůbec nezaujaly (viz Obr. 5.18), přestože pro ně byla predikována druhá a třetí fixace.

5.3.3 ModelBuilder

Ze všech vybraných VPL komponent má ModelBuilder jednotlivé prvky diagramů nejvýrazněji barevně rozlišené. Mezi základní prvky patří nástroje, které jsou reprezentovány žlutou barvou a tvarem obdélníku. Výstupní vrstvy jsou modré elipsy a výstupy zelené elipsy.



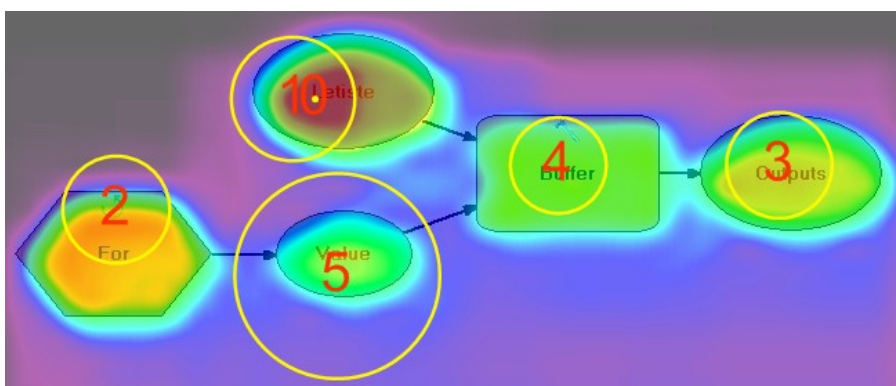
Obr. 5.19 Diagram *04_arcgis* – saliency mapa a pořadí sedmi fixací (ModelBuilder).



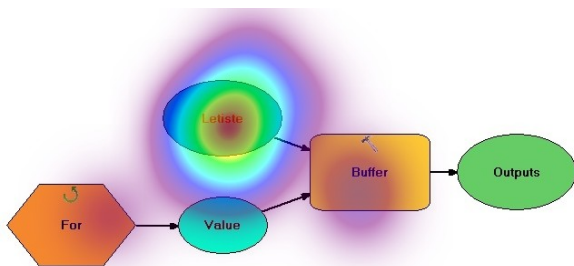
Obr. 5.20 Diagram *04_arcgis* – heatmap všech fixací (ModelBuilder).

U diagramu *04_arcgis* (Obr. 5.20) se shoduje místo s největší mírou nápadnosti (fixace 0) s oblastí, která nejvíce přitahovala pohledy respondentů. Jedná se o první vstup *SidlaPlochy*. Druhá předpokládané fixace byla vypočítána pro další vstupní vrstvu *Silnice* (Obr. 5.19), zde se ale respondenti zpravidla nedívali, obdobně dopadla třetí fixace (výstup). Prvky typu nástroj lze označit za nejméně nápadné prvky v diagramu, pokud nejsou brány v úvahu spojovací šipky.

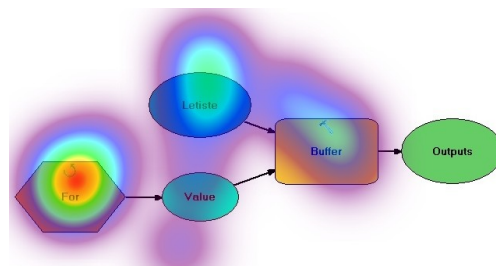
Další vybraný diagram *10_arcgis* (Obr. 5.22) lze označit za výjimečný tím, že u něho byly první dvě fixace predikovány do oblasti vstupní vrstvy *Letiste* (Obr. 5.21). Zde se také poprvé dívali respondenti testování metodou eye-tracking (viz Obr. 5.22). Druhá největší salience byla spočítána pro prvek typu iterátor reprezentovaný oranžovou barvou a tvarem šestiúhelníku. Tento prvek také přitahoval pohledy respondentů, stal se tak druhou oblastí zájmu (viz Obr. 5.23).



Obr. 5.21 Diagram *10_arcgis* – saliency mapa a pořadí šesti fixací (ModelBuilder).



Obr. 5.22 Diagram *10_arcgis* – heatmapa druhé fixace (ModelBuilder).



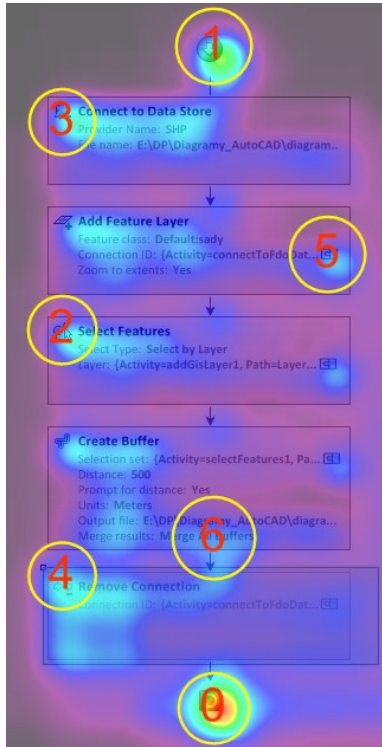
Obr. 5.23 Diagram *10_arcgis* – heatmapa třetí fixace (ModelBuilder).

5.3.4 Workflow Desinger

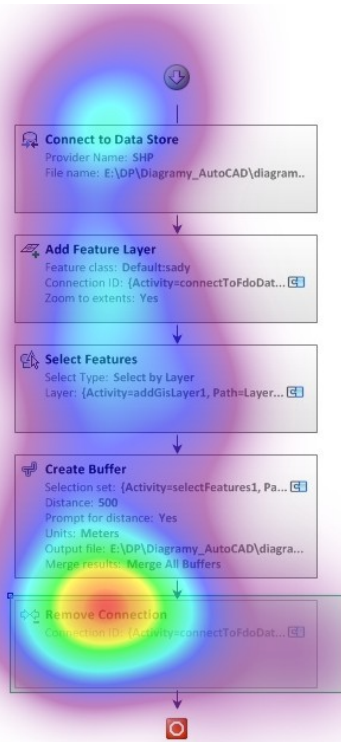
Poslední VPL komponenta Workflow Desinger je specifická tím, že v ní nelze změnit směr diagramu. Začátek diagramu reprezentovaný zeleným symbolem je vždy v horní části diagramu. Konec diagramu je v dolní části a zpravidla v jeho místě byla vypočítána největší salience (fixace 0). Druhá predikovaná fixace pak pro začátek diagramu (viz Obr. 5.24).

V diagramu *04_autocad* (Obr. 5.26) pozornost respondentů nejvíce přitahoval vypnutý prvek reprezentovaný světle zeleným obdélníkem (viz Obr. 5.25). Následně

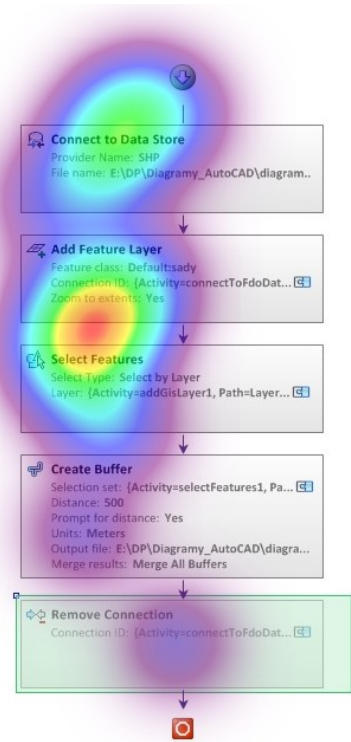
jejich pohled směřoval do horní části diagramu, aby mohli začít číst diagram od začátku obdobně jako u všech ostatních stimulů. Zajímavý byl výpočet salience pro paralelní diagram *07_autocad* (Obr. 5.28), kdy prvních pět fixací bylo predikováno do oblasti tmavě zeleného rámu (Obr. 5.27). Toto místo, ale vůbec neupoutalo pohledy respondentů, ti se opět snažili přečíst popis jednotlivých objektů v horní části diagramu (viz Obr. 5.28).



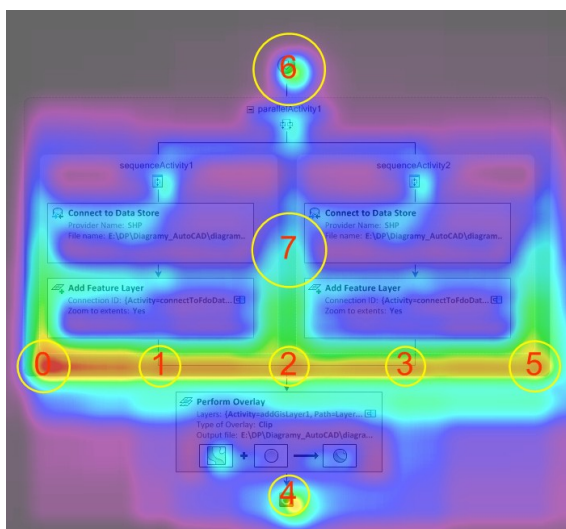
Obr. 5.24 Diagram *04_autocad* – saliency mapa a pořadí sedmi fixací (Workflow Designer).



Obr. 5.25 Diagram *04_autocad* – heatmapa všech fixací (Workflow Designer).



Obr. 5.26 Diagram *04_autocad* – heatmapa druhé fixace (Workflow Designer).



Obr. 5.27 Diagram *07_autocad* – saliency mapa a pořadí osmi fixací (Workflow Designer).



Obr. 5.28 Diagram *07_autocad* – heatmapa všech fixací (Workflow Designer).

5.3.5 Shrnutí

Z naměřených dat metodou eye-tracking bylo zjištěno, že uživatelé jsou zpravidla zvyklí číst diagramy směrem od vstupních vrstev (začátek diagramu) k výstupním vrstvám, což se potvrdilo u všech VPL komponent. Sledování experimentálních stimulů bylo tedy daleko více ovlivněno způsobem čtení diagramu než samotnou podobou prvků diagramů. Nelze proto zanedbat uspořádání jednotlivých prvků v diagramech. Diagramy bývají nejčastěji logicky uspořádány shora dolů nebo zprava doleva. Záměrně byly zařazeny do testování i diagramy, které měly vstup v dolní části diagramu. Rovněž u těchto diagramů bylo potvrzeno, že respondenti preferují hledání začátku diagramu (vstupní vrstvy) před nápadnosti prvků.

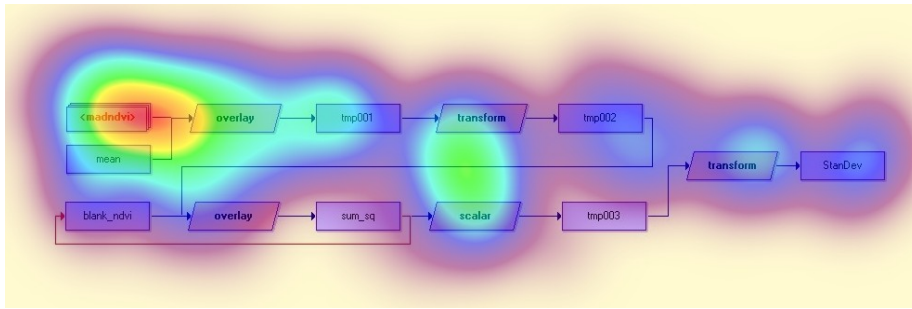
Po analýze všech 48 diagramů lze konstatovat, že vypočítaná salience v programu OGAMA a naměřená salience metodou eye-tracking se nejvíce shodovala u VPL komponenty ModelBuilder, zejména první fixace. Lze tedy předpokládat, že orientace v diagramech této VPL komponenty je pro uživatelé jednodušší než u ostatních VPL komponent. Testování diagramů datových toků metodou eye-tracking bylo z velké části ovlivněno směrem čtení diagramu jednotlivými respondenty nikoliv nápadnosti prvků.

5.4 Srovnání výsledků eye-tracking testování podle zkušeností s VPL

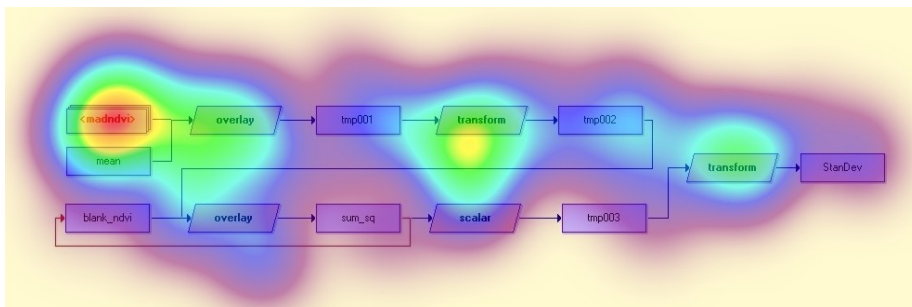
Posledním krokem této části práce bylo porovnání naměřených dat z testování salience metodou eye-tracking respondentů se zkušenostmi s VPL komponentami a bez zkušeností s vizuálními programovacími jazyky. Testování respondentů se znalostmi VPL komponent a bez zkušeností s VPL probíhalo v rámci samostatných eye-tracking experimentů. Z tohoto důvodu nebylo možno jednoduchou cestou porovnat scanpathy jednotlivých respondentů. Proto byly naměřená data vizualizována metodou heatmap. Vzhledem na rozsah práce nebylo možno se tomuto tématu věnovat ve větší míře.

Cílem tohoto srovnání bylo zjistit, jestli respondenty bez zkušeností s VPL bude v menší míře ovlivňovat způsob čtení diagramů a zdali budou věnovat více pozornosti nápadnosti jednotlivých prvků diagramů. Při porovnání druhé (případně třetí) fixace všech experimentálních stimulů bylo zjištěno, že respondenti bez zkušeností s VPL komponentami postupují při čtení diagramů datových toků obdobně jako respondenti, kteří mají znalosti VPL komponent (viz Obr. 5.29 a Obr. 5.30). Předpoklad tedy nebyl potvrzen a respondenti bez znalostí VPL byli rovněž při testování salience metodou eye-tracking ovlivněni způsobem čtení diagramů.

U některých experimentálních stimulů bylo zjištěno, že respondenti bez zkušeností s VPL čtou diagramy pomaleji. Toto tvrzení je zjevné zejména u diagramů datových toků VPL komponenty Workflow Designer, kde jednotlivé objekty obsahují více popisu než diagramy ostatních VPL komponent (viz Obr. 5.31 a 5.32).



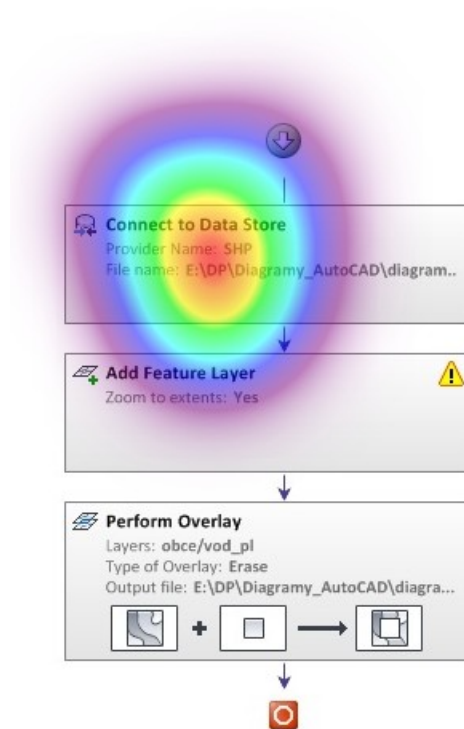
Obr. 5.29 Diagram 11_idrisi – heatmapa všech fixací, respondenti se zkušenostmi s VPL (Macro Modeler).



Obr. 5.30 Diagram 11_idrisi – heatmapa všech fixací, respondenti bez zkušeností s VPL (Macro Modeler).



Obr. 5.31 Diagram 06_autocad – heatmapa druhé fixace, respondenti se zkušenostmi s VPL (Workflow Designer).



Obr. 5.32 Diagram 06_autocad – heatmapa druhé fixace, respondenti bez zkušeností s VPL (Workflow Designer).

6 TESTOVÁNÍ INTERAKTIVNÍ PRÁCE METODOU EYE-TRACKING

Jedním z cílů této práce bylo zhodnotit interaktivní práci ve vizuálních programovacích jazycích používaných v GIS pomocí metody eye-tracking. Vzhledem k časově náročné analýze výsledků a znalostí respondentů byla k testování interaktivní práce vybrána pouze jedna VPL komponenta ModelBuilder. Připraveny byly postupně dva interaktivní testy, které probíhaly v prostředí programu ArcMap.

Pro studie použitelnosti není potřeba velké množství respondentů. Dostačující počet je již 5 respondentů (Nielsen, 2012). Je zřejmé, že při nulovém počtu testovaných, nelze získat žádné informace. Ale už z výsledků jednoho uživatele lze zjistit téměř třetinu všech potřebných poznatků (Nielsen, 2010). U sledování druhého respondenta se některé úkony shodují, ale je velmi pravděpodobné, že některé činnosti provedou uživatelé rozdílně. Avšak množství nových poznatků je nižší než jaké bylo zjištěno u prvního respondenta. S rostoucím počtem testovaných se činnosti jednotlivých respondentů stále častěji opakují a nových poznatků je méně a méně. Se zvyšujícím se počtem respondentů se tedy vytrácí efektivita testování.

První test interaktivní práce ve VPL komponentě ModelBuilder byl realizován v prosinci roku 2013. Dotestování proběhlo v lednu roku 2014. Testování se účastnili studenti 1. ročníku magisterského oboru Geoinformatika z Katedry geoinformatiky Univerzity Palackého v Olomouci v rámci předmětu KGI/DAMIN Data mining. Celkem bylo otestováno 10 respondentů.

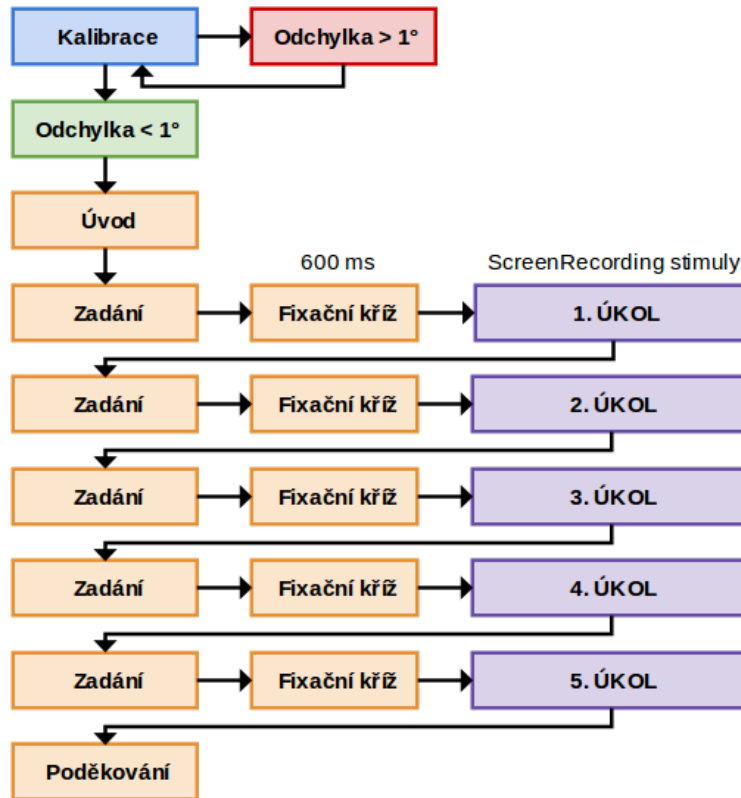
Druhé testování interaktivní práce proběhlo v letním semestru akademického roku 2013/2014 v rámci předmětu KGI/SKRI Skriptování v GIS. Účastníky byli studenti 2. ročníku bakalářského oboru Geoinformatika a geografie z Katedry geoinformatiky. Testování se celkem zúčastnilo 12 respondentů.

6.1 Struktura experimentů

Pro testování interaktivní práce ve vybrané VPL komponentě ModelBuilder byly připraveny dva experimenty se shodnou strukturou (viz Obr. 6.1). Oba experimenty obsahovaly pět úkolů. Každý test tedy obsahoval pět experimentálních stimulů ve formě snímání obrazovky tzv. ScreenRecording stimuly (viz Obr. 6.1), během kterých docházelo k nahrávání obrazovky a pohybu respondentova oka. Současně byly zaznamenány kliky myši či použití klávesnice.

Zadané úkoly nebyly časově omezeny. Na začátku testu byl vždy respondent upozorněn, že pokud nebude vědět, jak úkol vypracovat, může pokračovat dalším úkolem. Tímto se mělo předejít příliš dlouhému vypracovávání jednotlivých úkolů bez správného řešení.

Po úspěšné kalibraci se respondentovi zobrazila uvítací obrazovka, kde byl nabádán k pečlivému čtení zadání úkolů. Neboť při samotném vypracovávání jednotlivých úkolů již nebylo možné vrátit se zpět k zadání. Pokud si nemohl respondent při plnění úkolu vzpomenout, co bylo přesně v zadání, bylo mu ústně zopakováno.

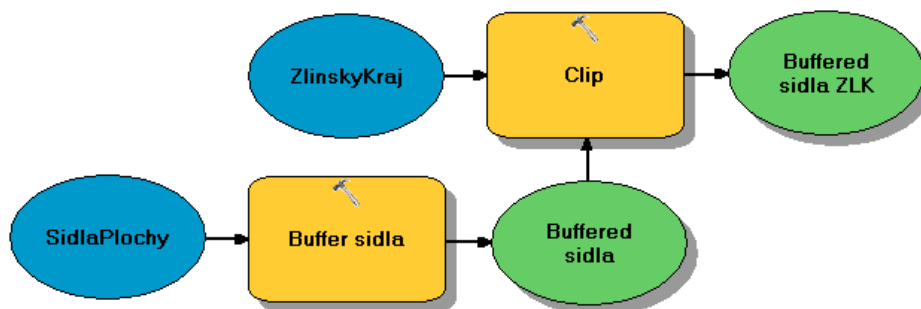


Obr. 6.1 Struktura experimentu pro testování interaktivní práce metodou eye-tracking.

6.1.1 První testování

Snahou bylo vytvořit pět jednoduchých úkolů tak, aby se celková délka testu pohybovala přibližně kolem 15 minut. Na začátku experimentu byly zařazeny jednodušší úkoly. U prvních třech úkolů měl respondent upravovat předem připravené funkční diagramy (viz Obr. 6.2). U posledních dvou úkolů měl respondent sám vytvořit nový diagram. Diagramy pro všechny úkoly jsou uloženy v toolboxu *Diagramy_interaktivita.tbx* (Příloha 4).

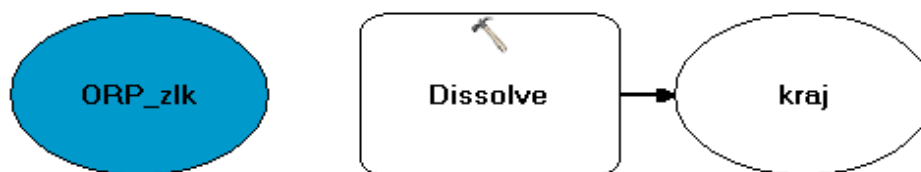
V zadání jednotlivých úkolů měli na závěr respondenti vždy napsáno jestli mají vyřešený úkol ukládat či nikoliv, případně zda výsledný diagram spustit (viz Příloha 2, Tab. 1). A dále jim bylo sděleno, ať minimalizují okno *ModelBuilder*, což bylo potřeba pro další bezproblémový průběh testu.



Obr. 6.2 Připravený diagram pro 2. úkol prvního testování interaktivní práce (ModelBuilder).

6.1.2 Druhé testování

Druhé testování interaktivní práce obsahovalo opět pět úkolů. Zase se postupovalo od jednodušších úkolů ke složitějším. Druhý úkol obsahoval předem připravený funkční diagram, který měl být rozšířen o další prvky. U prvního, třetího a čtvrtého úkolu bylo účelem předejít zdoluhavému hledání a přidávání prvků do diagramu. Proto byly pro tyto úkoly předem připraveny prvky, které měly být následně spojeny či upraveny tak, aby vznikl funkční diagram (viz Obr. 6.3). Poslední úkol se zabýval vytvořením nového diagramu (viz Příloha 2, Tab. 2). Diagramy pro jednotlivé úkoly jsou uloženy v toolboxu *Diagramy_interaktivita2.tbx* (Příloha 4).



Obr. 6.3 Připravené prvky pro 1. úkol druhého testování interaktivní práce (ModelBuilder).

6.2 Průběh testování

Průběh testování interaktivní práce metodou eye-tracking s experimentálními stimuly ve formě snímání obrazovky je mnohem náročnější než testování se stimuly typu obrázků. Kromě samotného experimentu je potřeba před každým spuštěním testu nachystat pracovní prostředí, ve kterém budou respondenti vykonávat zadané úkoly. Nejdůležitější je, aby toto prostředí měl každý respondent připraven stejně.

Pro předem připravené diagramy stačilo mít nachystaná minimalizovaná okna *ModelBuilder*, ale pro tvorbu nových modelů bylo potřeba připravit aplikaci ArcMap s otevřeným oknem *ArcToolbox* a oknem *Table Of Contents* s danými vrstvami. Podle toho kolik měli respondenti vytvářet nových modelů, tolik bylo spuštěno projektů v aplikaci ArcMap. Předem připravené diagramy respondenti zpravidla neukládali proto bylo jejich vrácení do původního stavu jednoduché. Nejtěžší bylo připravit stejné výchozí podmínky pro respondenty v připravené aplikaci ArcMap. Bylo nutné mimo jiné zavřít všechny rozbalené toolboxy v okně *ArcToolbox* či smazat hledaná slova v okně *Search*.

Testování nebylo náročné nejen pro respondenty, ale i pro operátora řídicího testování. Bylo potřeba pozorně sledovat respondenty, jak plní úkoly a v případě potřeby zopakovat zadání. Ale zejména, když četl respondent zadání, bylo nutné sledovat, o který úkol jde a podle toho maximalizovat připravené minimalizované okno pro daný úkol.

6.3 Hodnocení interaktivní práce

Během testování interaktivní práce metodou eye-tracking probíhalo snímání obrazovky a pohybu respondentova oka. Zároveň byly zaznamenány kliky myši či použití klávesnice. Výsledkem interaktivního testování byla videa (*.avi) s překryvným symbolem představující pohyb oka, která lze prohlížet v běžném multimediálním přehrávači. Výsledky lze také zobrazit v programu SMI BeGaze™.

Analýza výsledků byla časově náročná. Bylo potřeba zhlédnout všechny videa a zaznamenat úkony respondentů a také zjistit časy jednotlivých činností. Pro oba interaktivní testy byly vytvořeny přehledové tabulky úkonů, které vykonávali jednotliví respondenti (Příloha 7 a Příloha 8). Následně byly sestaveny tabulky časů pro vybrané úkony prováděné v jednotlivých úkolech (Příloha 2). Pro zvolené operace byly vytvořeny grafy (Příloha 1). Nakonec byly všechny úkoly opatřeny slovním komentářem.

6.3.1 První testování

Z prvního testování interaktivní práce ve VPL komponentě ModelBuilder metodou eye-tracking byly vyhodnocovány naměřená data od všech 10 respondentů.

Průměrná celková délka testu včetně kalibrace byla deset minut. Nejrychlejší byl respondent P05 s časem pět minut a 17 sekund. Nejdéle trval test respondentovi P11, který měl čas 14 minut a pět sekund. Průměrné časy vypracování jednotlivých úkolů narůstají, byl zde tedy naplněn záměr postupovat od jednodušších úkolů po složitější. Časově nejnáročnější je úkol č. 5, u kterého čas vypracování neklesl devíti respondentům z desíti pod dvě minuty a 20 sekund (viz Příloha 1, Obr. 1).

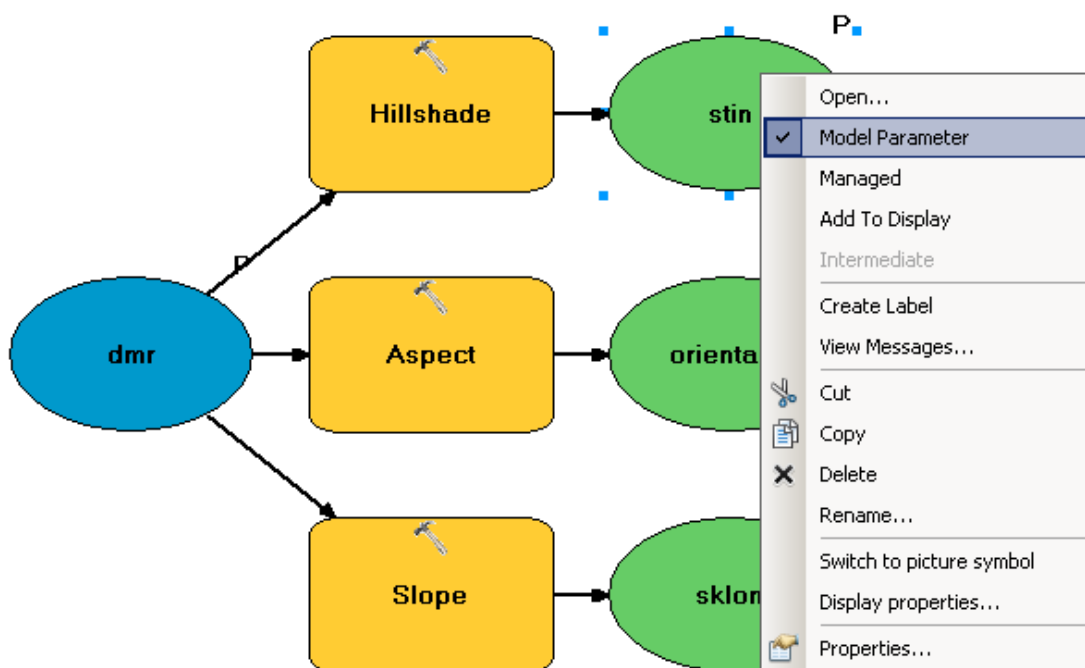
Po splnění prvních tří úkolů měli respondenti minimalizovat okno *ModelBuilder* pro další bezproblémové plnění následujících úkolů. S minimalizováním okna měl potíže respondent P07, který místo minimalizování okna maximalizoval, jak u 1. úkolu, tak u 2. úkolu. U úkolu č. 3 pro změnu okno ani neminimalizoval ani nemaximalizoval, z toho důvodu nemohly být zajištěny u tohoto respondenta stejné výchozí podmínky pro plnění 4. úkolu jako u ostatních respondentů. Pro vytvoření nového modelu u úkolu č. 4 byla pro respondenty připravena aplikace ArcMap s otevřeným oknem *ArcToolbox*. Respondent P07 si musel tyto podmínky nachystat sám, to znamená minimalizovat okna *ModelBuilder* z předcházejících úkolů. Časově tato příprava zabrala respondentovi celkem šest sekund z času plnění 5. úkolu a byla brána v úvahu při analýze získaných dat.

1. úkol 1. testování

Respondenti měli za úkol v připraveném diagramu změnit vstup a výstupy na parametrický vstup a výstupy. Správný postup byl jednoduchý. Stačilo postupně pravým tlačítkem myši kliknout na dané prvky a zvolit možnost *Model Parameter* (viz Obr. 6.4).

Úkol splnilo 90 % respondentů. Jeden respondent splnil úkol částečně, na parametr změnil pouze vstup (viz Příloha 1, Obr. 2). Lze to přičíst neznalosti, jak sám respondent uvedl při plnění úkolu. Ale je překvapivé, že pokud uměl změnit vstup na parametr, u výstupů se mu to nepovedlo, dokonce i když pravým tlačítkem vybral výstup *stin* a spatřil volbu *Model Parameter*.

Po analýze všech získaných dat bylo zjištěno, že všichni respondenti postupovali při plnění úkolu stejně. Nejdříve změnili na parametr vstup a až poté upravovali výstupy, ty měnili od shora dolů (tedy v pořadí *stin*, *orientace* a *sklon*). Všem devíti respondentům, kteří úspěšně splnili úkol, se povedlo změnit všechny čtyři prvky na parametrické do půl minuty (viz Příloha 2, Tab. 3). Z toho vyplývá, že respondenti tento jednoduchý úkon dobře ovládají.



Obr. 6.4 Změna výstupu na parametrický výstup pomocí *Model Parameter* (ModelBuilder).

2. úkol 1. testování

Druhým úkolem bylo automaticky zarovnat prvky v diagramu. Pro splnění stačilo pouze jedno kliknutí neboli použití tlačítka *Auto Layout* z nástrojové lišty. Proto lze úkol hodnotit buď jako splněný či nesplněný, nikoliv jako částečně splněný. Úkol splnilo 80 % respondentů a dva účastníci úlohu nesplnili (viz Příloha 1, Obr. 3).

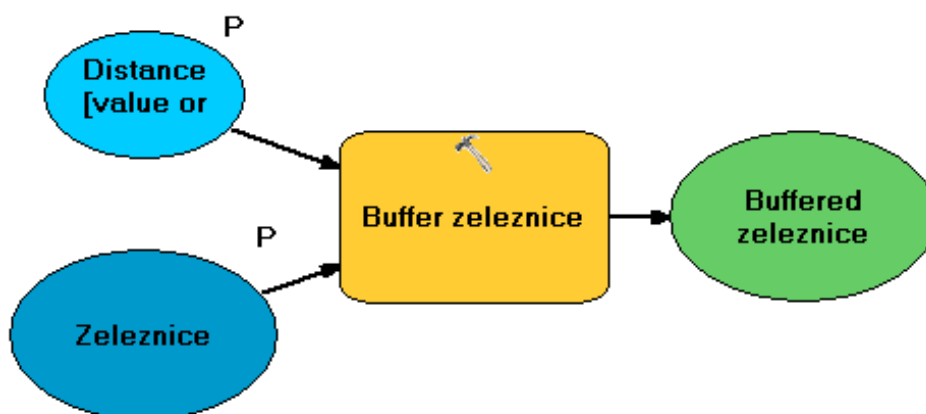
Většina respondentů (70 %) použila tlačítko *Auto Layout* z nástrojové lišty. Pouze jeden účastník testování využil nabídku *View*, z které pak vybral možnost *Auto Layout*. Jeden respondent před použitím správného tlačítka použil tlačítko *Full Extent* nacházející se v nástrojové liště vedle tlačítka *Auto Layout*. Čas, kdy bylo použito správné tlačítko nepřesáhl 15 sekund (viz Příloha 1, Obr. 3) až na jeden výjimečný případ (P11), který nevěděl, kde tlačítko hledat, ale nakonec ho našel.

Respondent P03, který nesplnil úkol, manuálně přeskládal prvky v diagram. Druhý respondent, který nesplnil úkol, rozbaloval nabídky, ale nemohl najít správné tlačítko a to ani poté, co ho jednou spatřil (viz Příloha 2, Tab. 4). Respondent si nebyl jistý, co má dělat, zadání mu bylo několikrát upřesňováno. Nakonec respondent sám řekl, že neví jak úkol vypracovat a pokračoval dalším úkolem.

3. úkol 1. testování

Třetí úloha už byla složitější než předchozí dva úkoly. Průměrná doba řešení úkolu byla jedna minuta. Respondenti měli do připraveného diagramu přidat další prvek: vzdálenost bufferu nastavit jako parametrickou proměnnou. Nejjednodušší postup byl následující: pravým tlačítkem myši kliknout na nástroj *Buffer zeleeznice* a vybrat možnosti *Make Variable* → *From Parameter* → *Distance [value or field]* (Obr. 6.6). Poté změnit vytvořenou proměnnou na parametrickou, pravým tlačítkem myši kliknout na proměnnou a zvolit volbu *Model Parameter*. Tato část úlohy byla obdobou 1. úkolu, kde se také pracovalo s parametrem (viz Obr. 6.4).

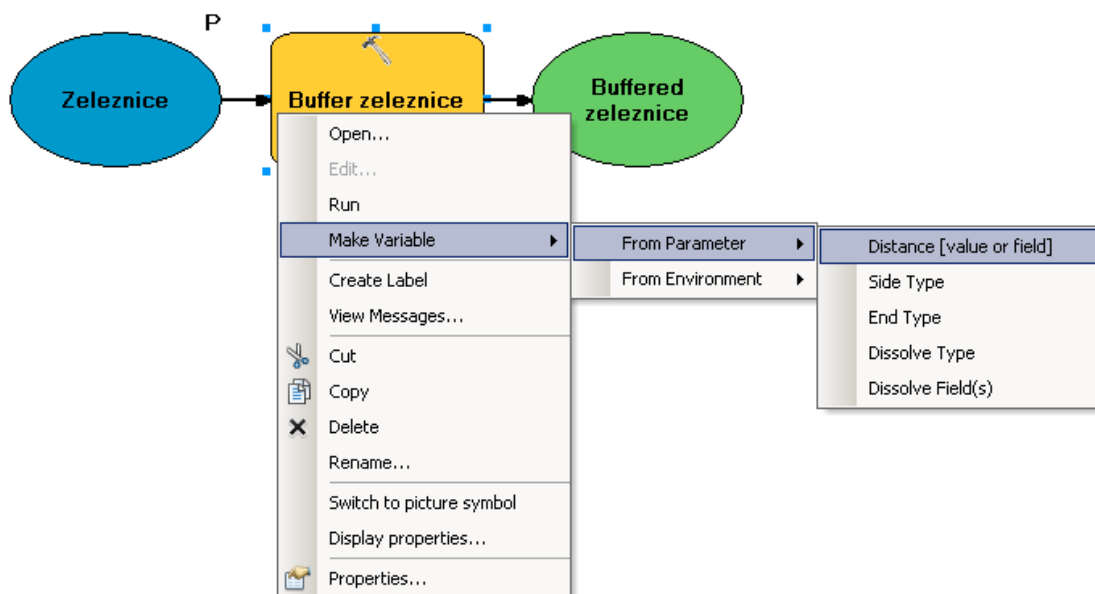
Správného řešení (viz Obr. 6.5) dosáhli pouze tři respondenti. Pět účastníků testu splnilo úkol částečně. Dva respondenti úlohu vyřešili špatně (viz Příloha 1, Obr. 4).



Obr. 6.5 Správné řešení 3. úkolu prvního testování interaktivní práce (ModelBuilder).

Respondent P08 se snažil nejdříve přidat proměnnou pomocí nabídky *Insert*, kde zvolil možnost *Create Variable*. Otevřelo se nové dialogové okno *Create Variable* pro výběr datového typu proměnné. Respondent vybral jako datový typ proměnné *Linear unit*. Následně pomocí spojovací šipky (nástroj *Connect* z nástrojové lišty) propojil proměnnou *Linear unit* a nástroj *Buffer zeleeznice*. Proměnnou nastavil jako parametrickou. Výsledkem byl ale nefunkční diagram, jelikož nebyla nastavena

vzdálenost bufferu v nově vytvořené proměnné *Linear unit*. Respondent po zjištění, že je model nefunkční, pracně vytvořenou proměnnou smazal. Dále postupoval jako všichni úspěšní řešitelé a vytvořil parametrickou proměnnou pomocí *Make Variable*. Po předcházejících úpravách zůstala v diagramu nenastavena vzdálenost bufferu a tím zůstal diagram stále nefunkční. Respondent při řešení úlohy obsáhl obě možné varianty tvorby proměnné, dvakrát vytvořil správný diagram. Bohužel ho, ale nenapadlo nastavit nějakou hodnotu pro vzdálenost bufferu, tím by se stal diagram funkční a respondentova práce by se dala považovat za splnění úkolu.



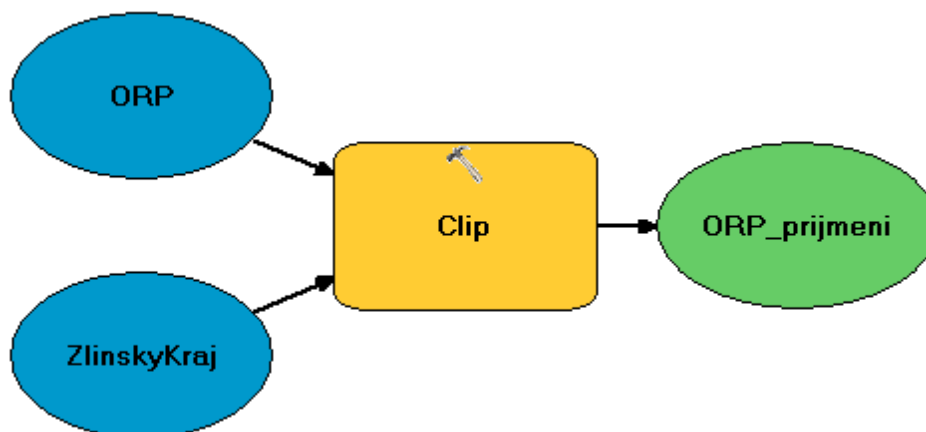
Obr. 6.6 Vzdálenost bufferu přidávána do diagramu jako proměnná pomocí *Make Variable* → *From Parameter* → *Distance [value or field]* (ModelBuilder).

Dalším respondentům, kteří splnili úkol částečně, chybělo ke splnění úlohy nastavení proměnné jako parametr. Jejich čas vykonávání úkolu byl pod 40 sekund, můžeme tedy předpokládat, že nastavení parametru nebrali v úvahu. Jedině respondent P03 měl čas delší, téměř minutu a půl. Tento respondent, ale stihl proměnnou vytvořit dvakrát, poprvé ji ale smazal, poté usoudil, že nic jiného nevymyslí, tak ji opětovně vytvořil (viz Příloha 2, Tab. 5). Parametr nevytvořil, ale to se u něj dalo předpokládat po neznalosti tvorby parametrů v první úloze. Respondent P02 jako jediný zarovnal diagram pomocí tlačítka *Auto Layout*. To bylo použito již v předcházejícím úkolu, takže bylo předpokládáno, že bude využito více respondenty. Tato hypotéza se, ale nepotvrdila.

Neúspěšní respondenti v plnění 3. úkolu přiznali, že nevěděli jak postupovat. Jeden účastník změnil výstup na parametr. Druhý se pokoušel přidat do diagramu iterátor, nakonec ho, ale smazal.

4. úkol 1. testování

Úkol č. 4 byl časově náročnější než první tři úlohy, jelikož respondenti měli sami vytvořit nový fungující diagram. Respondentům trvalo vypracovat úkol průměrně dvě minuty a 20 sekund. Úkolem bylo vytvořit nový model s názvem *4_prijmeni*. Následně sestavit diagram, který provede ořezání bodové vrstvy *ORP* polygonovou vrstvou *ZlinskyKraj* a výstupní vrstvu pojmenovat *ORP_prijmeni*. V předešlých úkolech se neměly diagramy ukládat, v této úloze se měl hotový diagram uložit a nechat proběhnout (*Run*).

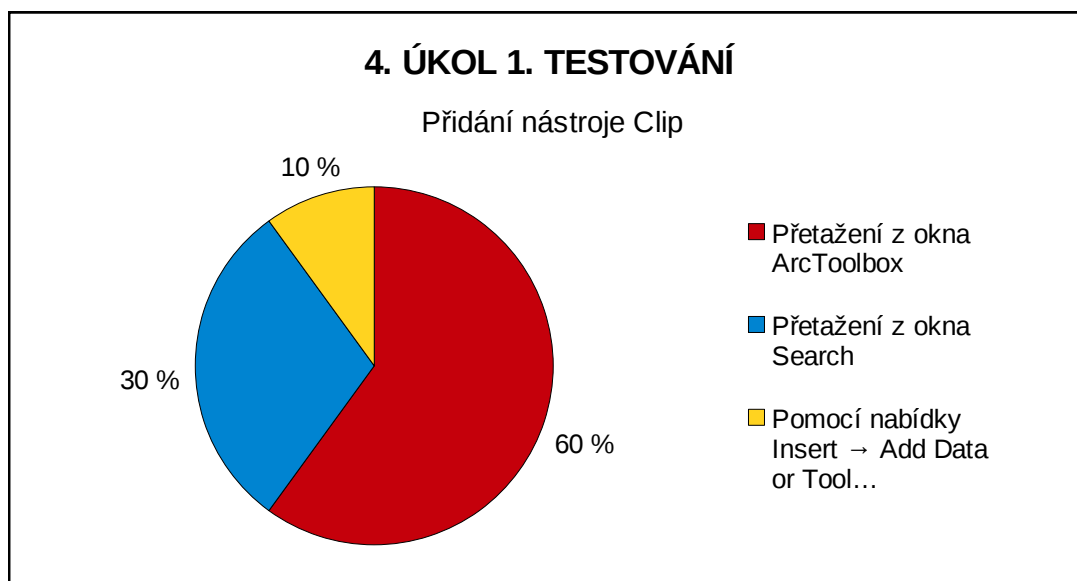


Obr. 6.7 Správné řešení 4. úkolu prvního testování interaktivní práce (ModelBuilder).

Všem respondentům se podařilo vytvořit nový model a sestavit diagram s nástrojem *Clip*, po jehož spuštění vznikla výstupní vrstva. Některým respondentům dělalo problém pojmenovat nově vytvořený model či výstupní vrstvu. Těchto respondentů bylo 50 % a jejich výsledek byl hodnocen jako částečně splněný úkol (viz Příloha 1, Obr. 5). Úspěšných řešitelů se správně pojmenovaným modelem (*4_prijmeni*) a výstupní vrstvou bylo pět (viz Obr. 6.7). Čas nejrychlejšího respondenta, který splnil úkol, byl jedna minuta a 41 sekund.

Při pojmenování nového modelu respondenti využili dva způsoby. Za prvé lze nový model uložit jako pomocí nabídky *Model* → *Save As*. Tato volba je dostupná v aplikaci ArcMap teprve od verze 10.1. Další způsob je následující: zavřít okno nově vytvořeného modelu a přejmenovat model v okně *ArcToolbox* (*Rename*) a následně znovu otevřít model pro editaci (*Edit*). První možnosti využilo pět respondentů, druhé pouze dva. U druhého způsobu přejmenování modelu bylo nutné přesně dodržet postup, což se nepovedlo třem respondentům. Jejich přejmenování modelu bylo tedy neúspěšné. Tito respondenti nezavřeli okno nového modelu, ale pouze model přejmenovali v okně *ArcToolbox* pomocí volby *Rename*. Nikdo si z nich nevšiml, že se jim nezměnil název otevřeného okna *ModelBuilder*. Všichni respondenti pojmenováním modelu zahájili vypracovávání úkolu, nikdo si nenechal tento úkon na závěr.

Dalším krokem bylo vytvoření vlastního diagramu (viz Obr. 6.7). Vstupní a ořezovou vrstvu přetáhli všichni respondenti z okna *Table Of Contents* až na respondenta P04, který přetáhl pouze vstupní vrstvu *ORP*. Přidání nástroje *Clip* bylo více různorodé. U respondentů se objevili tři postupy, jak přidat nástroj do diagramu (viz Obr. 6.8). Nejčastější formou bylo přetažení a to buď z okna *ArcToolbox* (6 respondentů), kde si nástroj sami vyhledali nebo z okna *Search* (3 respondenti), kde jim byl nástroj vyhledán po zadání klíčového slova. Všichni zvolili jako klíčové slovo *clip*. Dva respondenti se pokoušeli přetáhnout nástroj z nabídky *Geoprocessing* hlavního okna aplikace ArcMap.

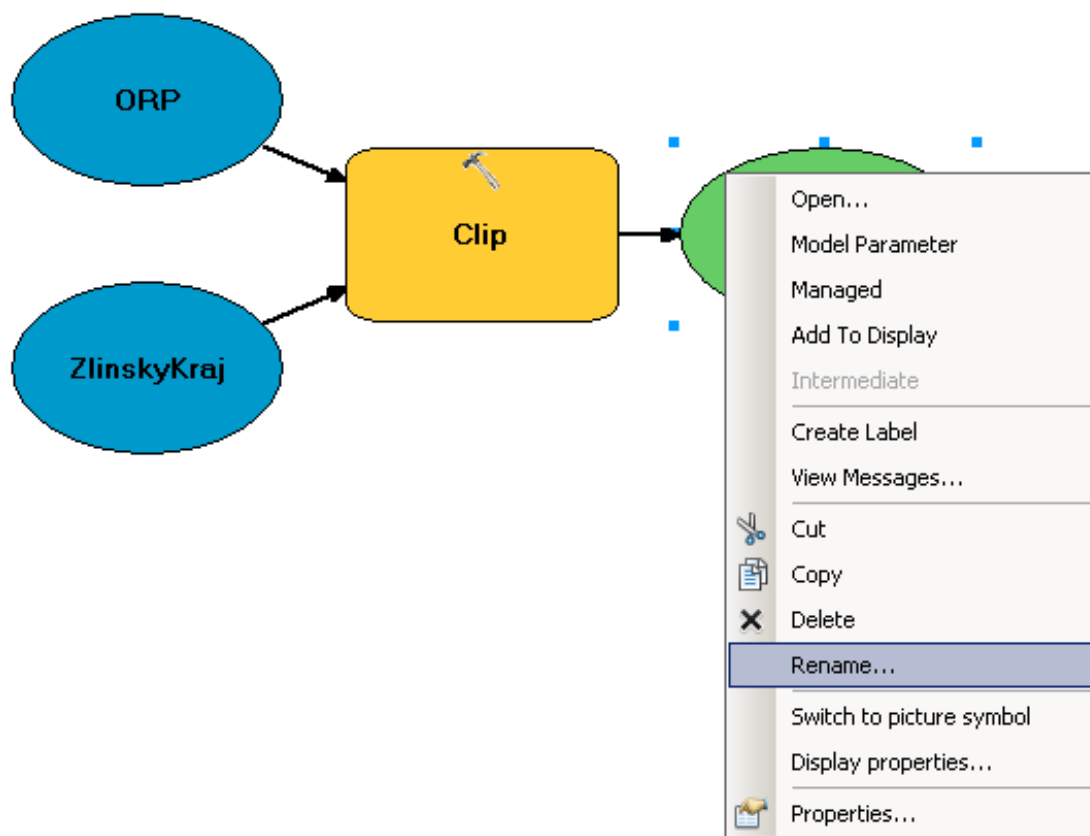


Obr. 6.8 Přidání nástroje *Clip* ve 4. úkolu prvního testování interaktivní práce (10 respondentů).

K propojení vstupní a ořezové vrstvy s nástrojem využilo 80 % respondentů spojovacích šipek (nástroj *Connect* z nástrojové lišty). Pouze dva respondenti dvojklikem otevřeli dialogové okno nástroje *Clip*, kde vybrali vstupní a ořezovou vrstvu. Měli na výběr jestli použijí vrstvy, které jsou v okně *Table Of Contents* (žlutý symbol) nebo již ty přetažené do okna *ModelBuilder* (modrý symbol). Bohužel vybrali první možnost, takže po potvrzení dialogového okna se do diagramu přidaly tyto vrstvy znova, ale už byly spojené s nástrojem. Následně museli v diagramu smazat vrstvy, které již předtím přetáhli z okna *Table Of Contents*. Jeden respondent měl potíže se správným propojením a nastavením vstupní a ořezové vrstvy.

Výstupní vrstva se vytvořila všem respondentům, ale u třech nastal problém s pojmenováním. Přejmenovali totiž pouze prvek v diagramu (viz Obr. 6.9) nikoliv výstupní vrstvu, která se následně vytvořila po proběhnutí diagramu. Změnit název výstupní vrstvy je potřeba v dialogovém okně výstupu (6 respondentů) či nástroje (1 respondent).

Na závěr využili tři účastníci testování tlačítko *Auto Layout* pro automatické zarovnání diagramu. Nakonec měli respondenti diagram nechat proběhnout. Čtyři respondenti před tímto úkonem provedli nejdříve validaci vytvořeného diagramu (*Validate Entire Model*).



Obr. 6.9 Přejmenování prvku v diagramu pomocí *Rename* (ModelBuilder).

5. úkol 1. testování

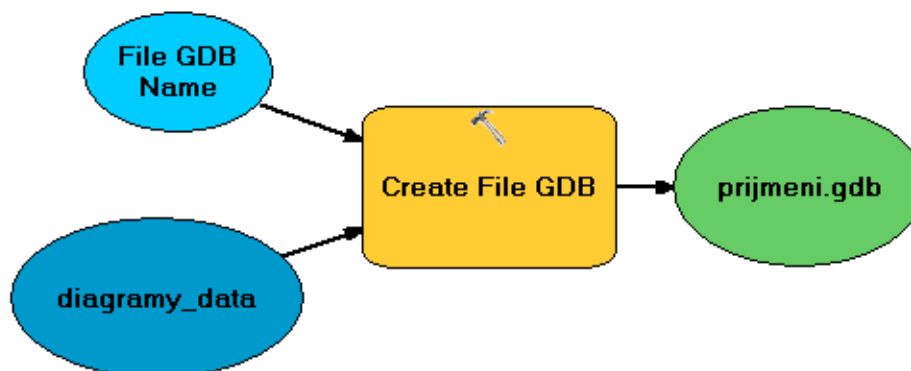
Posledním úkolem bylo vytvořit nový model s názvem *5_prijmeni*. Dále sestavit diagram pro vytvoření nové souborové geodatabáze s názvem *prijmeni.gdb*. Geodatabázi umístit do adresáře *diagramy_data* a nastavit název geodatabáze jako proměnnou (viz Obr. 6.10).

Úkol č. 5 byl z prvního testování časově nejnáročnější. Respondentům trval průměrně tři minuty a 14 sekund. Ke správnému řešení dospěli tři respondenti, pět jich zvládlo úlohu částečně a dva respondenti si nevěděli rady (viz Příloha 1, Obr. 6).

Vytvoření nového modelu a jeho následné pojmenování bylo obdobou 4. úkolu. Téměř všichni respondenti (90 %) v této části postupovali stejně jako v předcházející úloze. Pouze jednomu respondentovi ze tří, kteří v minulém úkole neuměli pojmenovat model, se podařilo zlepšit a tentokrát nový model správně přejmenovat.

U toho úkolu respondenti nepoužívali žádná vstupní data. Pro splnění úlohy bylo nejdůležitější najít samotný nástroj pro tvorbu souborové geodatabáze a nastavit

správně cestu k adresáři, kde se vytvoří. Pro respondenty bylo hledání méně známého nástroje *Create File GDB* obtížnější než hledání nástroje *Clip* u 4. úkolu. Nástroj *Create File GDB* se nachází v toolboxu *Data Management Tools* a v sadě nástrojů *Workspace*. Pokud respondenti použili k hledání nástroje okno *ArcToolbox* a rozbalili příslušný toolbox, většinou spatřili nejdříve sadu nástrojů *File Geodatabase* a zde se snažili najít odpovídající nástroj (viz Příloha 2, Tab. 7). Sady nástrojů jsou v toolboxech řazeny podle abecedy, tudíž sada nástrojů *Workspace* se nachází úplně dole. Z toho důvodu bylo obtížné ji najít a ještě k tomu otevřít, zejména pokud respondent nevěděl, kde hledat.

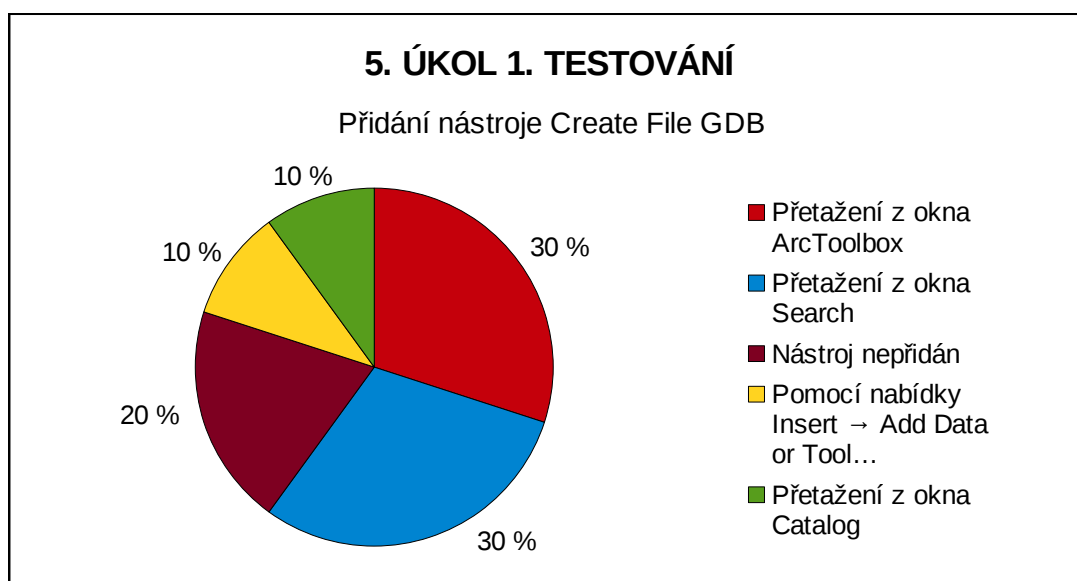


Obr. 6.10 Správné řešení 5. úkolu prvního testování interaktivní práce (ModelBuilder).

K postupům z předešlého úkolu (viz Obr. 6.8), jak přidat nástroj do diagramu, přibyla další možnost a to přetažení z okna *Catalog* (viz Obr. 6.11). Celkem šest respondentů použilo okno *Search* pro vyhledání daného nástroje, ale pouze tři respondenti z tohoto okna nástroj přetáhli. U dvou respondentů byla zjištěna neznalost možnosti přetažení nástroje z okna *Search*. Po rychlém vyhledání nástroje v okně *Search* se přemístili do okna *ArcToolbox*, kde znova pracně hledali nástroj podle adresářové cesty, která se zobrazuje k nalezeným nástrojům v okně *Search*. Poslední respondent, který použil při hledání nástroje okno *Search* a nepřetáhl z něho nástroj do diagramu, zvolil o něco rychlejší postup. Po kliknutí na adresářovou cestu nástroje *Create File GDB* v okně *Search* se rozbalil příslušný toolbox a sada nástrojů v okně *Catalog*, z kterého následně respondent nástroj přetáhl do diagramu.

Všichni respondenti, kteří splnili nebo částečně splnili úkol, správně nastavili cestu k adresáři, kde se má vytvořit nová geodatabáze. Z těchto osmi respondentů pouze jeden vytvořil proměnnou *File GDB Location* pomocí kliknutí pravého tlačítka myši na nástroj a následného vybrání volby *Make Variable* → *From Parameter* → *File GDB Location*. Ostatní nastavili příslušný adresář v dialogovém okně nástroje. Proměnnou *File GDB Name* se podařilo přidat do diagramu šesti respondentům. Dva respondenti měli problém s názvoslovím. Když totiž měli nastavit název geodatabáze jako proměnnou, pouze změnili výstup (*prijmeni.gdb*) na parametrický.

Respondent P08 zvládl úkol v pořádku, jen místo *prijmeni.gdb* pojmenoval geodatabázi *test.gdb*. Sám si to uvědomil, až při kontrole vytvořené geodatabáze v okně *Catalog*. Respondent P11 byl vytrvalý v plnění úkolu a nechtěl neúspěšně ukončit vypracovávání úkolu. Jeho čas byl oproti ostatním respondentům dvojnásobný (viz Příloha 1, Obr. 6). Opět jako v předešlém úkolu pouze přejmenoval výstupní prvek v diagramu místo pojmenování výstupní geodatabáze. Model byl tedy nefunkční a respondent si nevěděl rady, čím to je. Z důvodu dlouhé doby plnění úkolu byla respondentovi dána rada, ať zkusí v dialogovém okně nástroje přepsat název proměnné *File GDB Name*. Po tomto úkonu se stal diagram funkční a po spuštění proběhl v pořádku. Jelikož ke správnému závěru nedošel respondent sám, nelze brát úkol za splněný, ale pouze za částečně splněný.



Obr. 6.11 Přidání nástroje *Create File GDB* v 5. úkolu prvního testování interaktivní práce (10 respondentů).

6.3.2 Druhé testování

Druhého testování interaktivní práce metodou eye-tracking se celkem zúčastnilo 12 respondentů. U respondentů P02 a P03 nebyly vytvořeny shodné podmínky pro plnění 1. úkolu jako u ostatních respondentů. Z toho důvodu nebyli zahrnuti do analyzování naměřených dat. Celkem byla tedy vyhodnocována data od deseti respondentů, obdobně jako u prvního testování interaktivní práce.

U dvou respondentů P07 a P09 nastaly při testování technické potíže. Z tohoto důvodu muselo být testování těchto respondentů rozděleno na dvě části a při vyhodnocování byl brán P07 a P08 jako jeden respondent, taktéž P09 a P10 (viz Příloha 8). U testování respondenta P07 došlo během čtení zadání třetího úkolu a současného chystání okna *ModelBuilder* k neočekávanému ukončení celého programu ArcMap. Potíže přetrvávaly i u respondenta P09. V závěru druhém úkolu

při otevření dialogového okna nástroje vyskočilo varovné hlášení o chybě skriptu (viz 2. úkol 2. testování). Tyto technické problémy se u těchto dvou respondentů projevíly do celkového času vypracovávání testu a zejména pak do času 2. úkolu.

Celková průměrná délka testů v druhém testování interaktivní práce včetně kalibrace byla 11 minut a 49 sekund. Nejrychlejší respondent vypracoval test za osm minut a 15 sekund. Naopak nejpomalejší respondent měl více než dvakrát delší čas a to 18 minut a 22 sekund. Průměrný součet časů jednotlivých úkolů v tomto testování byl větší o 41 sekund než v prvním testování. V druhém testování byl průměrný součet časů úkolů osm minut a deset sekund, v prvním testování to bylo sedm minut a 29 sekund.

Obdobně jako u prvního testování měli respondenti po splnění prvních čtyř úkolů minimalizovat okno *ModelBuilder* pro další bezproblémový průběh testu. S tímto úkonem měl po splnění 1. úkolu obtíže jeden respondent (P13), který okno maximalizoval místo minimalizoval. Nemohly tedy být u úkolu č. 5 pro něj zajištěny shodné výchozí podmínky jako u ostatních respondentů. Příprava těchto podmínek zabrala sedm sekund času z plnění 5. úkolu, což bylo zohledněno při vyhodnocování získaných dat.

1. úkol 2. testování

V prvním úkolu měli respondenti upravit připravené prvky (viz Obr. 6.3). Vzniknout měl funkční diagram, který provede generalizaci (rozpuštění) hranic obcí na hranici kraje. Respondenti měli využít atribut *KRAJ* jako *Dissolve_Field(s)*, tedy pole, podle kterého dojde ke generalizaci hranic.

Respondenti vypracovali úlohu průměrně za 55 sekund. Správně úkol vyřešilo 50 % respondentů. Všem respondentům se povedlo propojit vstupní vrstvu s nástrojem *Dissolve*. K tomu bylo použito spojovací šipky (nástroj *Connect* z nástrojové lišty). Výjimkou byl pouze jeden respondent, který propojil vstup a nástroj přes dialogové okno nástroje.

Dalším krokem bylo správné nastavení *Dissolve_Field(s)* v dialogovém okně nástroje *Dissolve*. Potřeba bylo vybrat atribut *KRAJ* pro rozpuštění hranic obcí na hranici kraje. Tuto část nesplnilo 50 % respondentů, proto byl jejich výsledek hodnocen jako částečné splnění úkolu (viz Příloha 1, Obr. 8). Tři respondenti ani neotevřeli dialogové okno nástroje, kde mohli *Dissolve_Field(s)* nastavit (viz Příloha 2, Tab. 8).

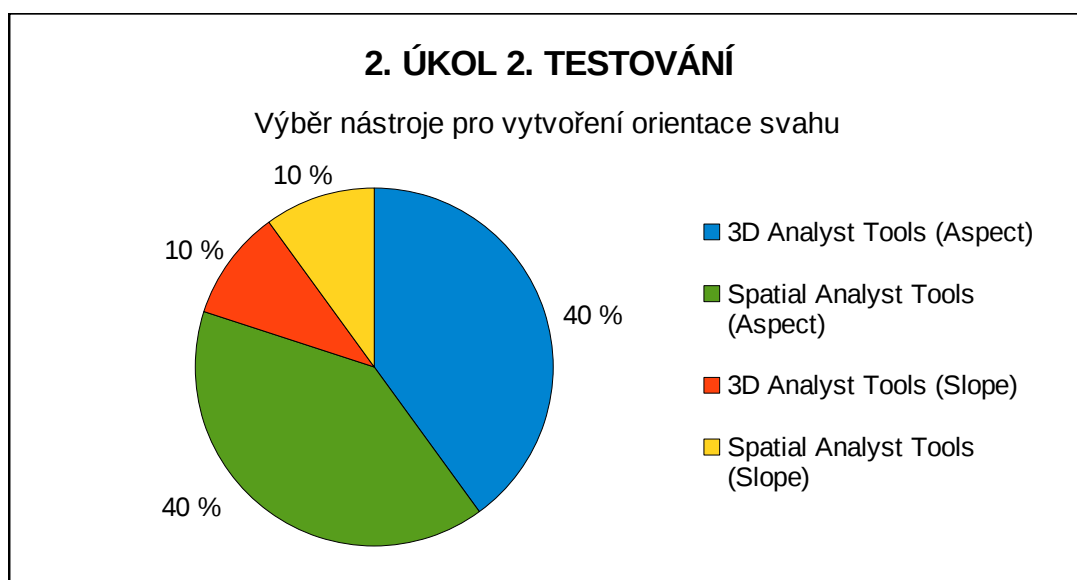
2. úkol 2. testování

Pro druhý úkol byl připraven funkční diagram pro vytvoření digitálního modelu reliéfu. Úkolem bylo přidat do diagramu nástroj, který vytvoří z vrstvy *DMR* vrstvu orientace svahu s názvem *orientace*.

Prvním krokem bylo vyhledání nástroje *Aspect*, následně jeho přidání do diagramu, spojení s již vytvořenou vrstvou *DMR* a správné pojmenování výstupu. Manuální

hledání nástroje v okně *ArcToolbox* preferovalo 70 % respondentů, ostatní použili okno *Search*. Dva respondenti mylně vybrali nástroj *Slope*, který slouží k vytvoření vrstvy sklonu svahu nikoliv orientace. Jejich výsledek byl hodnocen jako nesplnění úkolu (viz Příloha 1, Obr. 9). Respondenti měli možnost volby, nástroj mohli použít buď z toolboxu *3D Analyst Tools* nebo *Spatial Analyst Tools* (viz Obr. 6.12).

Nejdelší čas vypracování úkolu byl zjištěn u respondenta P09 (viz Příloha 2, Tab. 9). Jedním z důvodů dlouhé doby plnění úlohy je, že respondent nevěděl, jestli použít nástroj *Slope* (sklon) nebo *Aspect* (orientace). Využil proto volby *Item Description*, kde si přečetl popisy obou nástrojů. Následně vybral správný nástroj. Další příčinou nejdelšího času byly technické problémy, které nastaly při vypracování úlohy. Při snaze respondenta správně pojmenovat výstupní vrstvu vyskočilo při otvírání dialogového okna nástroje varovné hlášení o chybě skriptu. Z audio a video záznamu bylo zjištěno, že by respondent úlohu dokončil správně, proto je považován jeho úkol za splněný.



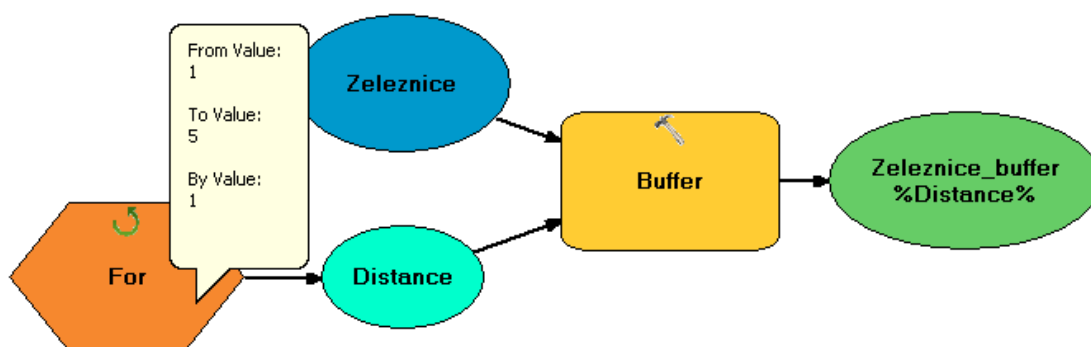
Obr. 6.12 Výběr nástroje pro vytvoření orientace svahu v 2. úkolu druhého testování interaktivní práce – správně nástroj *Aspect*, špatně nástroj *Slope* (10 respondentů).

Čtyřem respondentům by se po spuštění diagramu nevytvořila výstupní vrstva se správným názvem *orientace*, jelikož pouze přejmenovali prvek v diagramu (pomocí *Rename* viz Obr. 6.9). Toto je velmi častá chyba, kterou si respondenti ani neuvědomují, že ji dělají. Ostatní respondenti přejmenovávali výstupní vrstvu v dialogovém okně nástroje. Pouze jeden respondent z šesti využil dialogové okno výstupu. Těchto 60 % respondentů, kteří správně pojmenovali výstup, úspěšně zvládli zadaný úkol. Tři respondenti využili tlačítka *Auto Layout* k automatickému uspořádání diagramu.

3. úkol 2. testování

Respondenti měli za úkol sestavit z daných prvků diagram, který bude vytvářet kolem vstupní vrstvy obalové zóny (buffery) ve vzdálenosti 1, 2, 3, 4 a 5 kilometrů. Důležité bylo správně nastavit hodnoty v iterátoru *For* a spojit proměnnou *Distance* s nástrojem *Buffer* (Obr. 6.13). Rozhodnutí, jestli je potřeba do názvu výstupu přidat proměnnou *%Distance%*, aby se výstupní vrstva nepřepisovala, bylo ponecháno na samostatné úvaze respondenta.

Prvním krokem při plnění úkolu bylo zpravidla propojení vstupní vrstvy *Zeleznice* a nástroje *Buffer*. Tento úkon provedlo 90 % respondentů (Příloha 2, Tab. 10). Následovala snaha propojit iterátor s jakýmkoliv prvkem v diagramu. Třem respondentům se povedlo propojit proměnnou *Distance* s nástrojem *Buffer*, ale pouze dva respondenti věděli, co dělají, třetí respondent P09(10) jen zkoušel propojovat prvky stylem pokus omyl a následně správné spojení stejně smazal.



Obr. 6.13 Nastavené hodnoty v iterátoru *For* ve splněném úkolu č. 3 druhého testování interaktivní práce (ModelBuilder).

Dále se respondenti věnovali nastavení hodnot v iterátoru *For*. O zadání hodnot se pokoušelo sedm respondentů, ale jenom tři respondenti nastavili s ohledem na jednotky hodnoty správně. Do iterátoru se zadávaly tři hodnoty: od které hodnoty se má začít počítat, u které hodnoty se má skončit, a jak velký má být krok. Hodnoty se nastavovaly podle jednotek v nástroji *Buffer*. Výchozí jednotky byly metry, v kterých hodnoty nastavili všichni tři respondenti. Druhou možností bylo změnit nastavení jednotek v nástroji na kilometry a zadat hodnoty jako na Obr. 6.13.

Tři respondenti začali uvažovat nad pojmenováním výstupu. Správně jim došlo, že výstup se bude přepisovat, proto do názvu výstupu připsali proměnnou *%Distance%*, která zaručí vytvoření více výstupních vrstev.

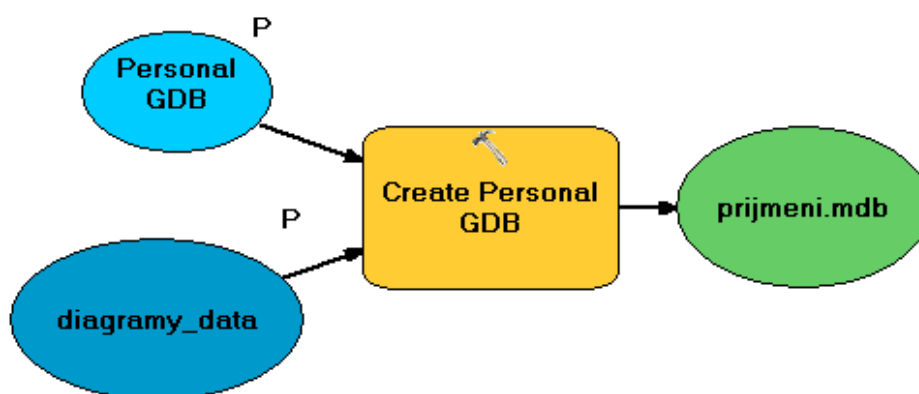
Splnit úkol se povedlo pouze respondentovi P14 (Příloha 1, Obr. 10), ten jediný si pohlídal všechny nástrahy, které sestavování diagramu skrývalo. Úlohu nesplnilo 60 % respondentů a byli to ti, kteří při řešení úkolu prohlásili, že neví, jak postupovat. Třem respondentům se podařilo splnit úkol částečně, ke splnění jim zpravidla chyběl jeden či dva úkony. U respondenta P06 to bylo propojení proměnné *Distance*

a nástroje *Buffer* a u respondenta P07(08) správné nastavení jednotek, hodnoty totiž zadával v kilometrech a nastaveny měl metry. Respondent P11 nespojil proměnnou *Distance* a nástroj *Buffer* a navíc neuvažoval nad přepsáním výstupu.

Po analýze této úlohy lze konstatovat, že zadaný úkol byl pro respondenty příliš náročný. Studenti již měli zkušenosti s iterátorem, ale jeho zapojení do diagramu bylo pro ně i přesto velmi obtížné. Největším omylem bylo, že se snažili propojit šestiúhelníkový prvek (iterátor *For*) s dalšími prvky. Místo toho bylo potřeba spojit proměnnou *Distance*, která na iterátor navazovala, s nástrojem *Buffer* (viz Obr. 6.13).

4. úkol 2. testování

Tento úkol byl obdobou 5. úkolu prvního testování (viz Obr. 6.10). Rozdíl spočíval v tom, že v tomto případě již byl připravený nástroj pro tvorbu osobní geodatabáze (*Create Personal GDB*), respondenti se tak vyhnuli zdlouhavému hledání nástroje. Navíc ještě měli respondenti změnit nově přidané proměnné na parametrické. Úkolem tedy bylo k připravenému nástroji přidat další dva prvky a to umístění osobní geodatabáze (*Personal GDB Location*) a její název (*Personal GDB Name*) jako parametrické proměnné (viz Obr. 6.14). Umístění mělo být nasměřováno do adresáře *diagramy_data* a název nové geodatabáze měl být *prijmeni.mdb*. Pro následnou kontrolu měl být diagram spuštěn (*Run*), ale změny neměl respondent ukládat.



Obr. 6.14 Správné řešení 4. úkolu druhého testování interaktivní práce (ModelBuilder).

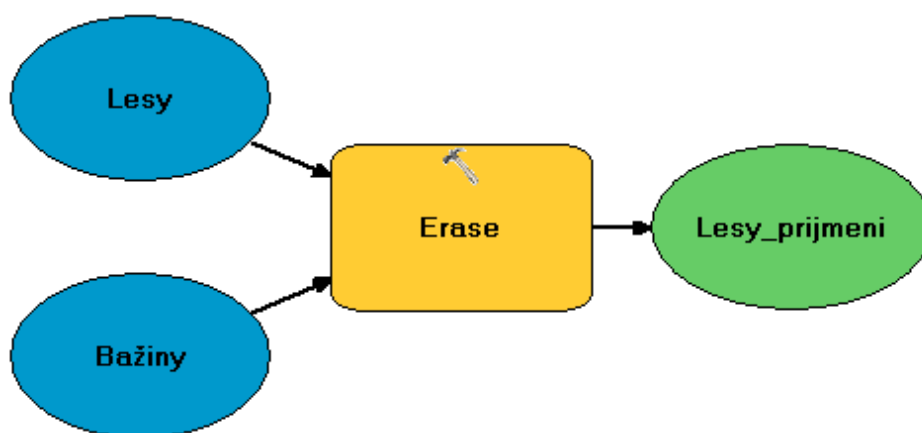
Úkol se opět povedlo úspěšně vyřešit pouze jednomu respondentovi z deseti. Ostatní respondenti splnili úkol částečně. Vytvořit novou osobní geodatabázi se správným názvem se povedlo devíti respondentům. Respondentovi P11 se sice povedlo přidat obě požadované proměnné (Obr. 6.14) a nastavit je jako parametrické. Ale výsledný diagram byl přesto nefunkční a to z důvodu nezadání názvu pro novou geodatabázi, pouze byl přejmenován prvek výstupu pomocí *Rename* (viz Obr. 6.9).

Všem respondentům se povedlo přidat do diagramu proměnnou *Personal GDB Location* a nastavit cestu ke správnému adresáři pomocí dialogového okna nástroje. Pět respondentů změnilo tuto proměnnou na parametrickou. Problémem byla druhá proměnná *Personal GDB Name*, kterou přidali jen tři účastníci testování a poté ji na

parametrickou nastavili dva (Příloha 2, Tab. 11). Jiní dva respondenti změnili na parametr výstup. Je zajímavé, že jim nedošlo, že nemění na parametr nově přidaný prvek, jak bylo v zadání. Stejně postupovali i dva respondenti u 5. úkolu prvního testování. K přidání druhé proměnné bylo potřeba využít možnosti *Make Variable* → *From Parameter* → *Personal GDB Name* obdobně jako v 3. a 5. úkolu prvního testování (viz Obr. 6.6).

5. úkol 2. testování

Posledním úkolem bylo vytvořit nový model s názvem *prijmeni*. Následně sestavit diagram, který provede vymazání (nástroj *Erase*) polygonové vrstvy *Bažiny* z polygonové vrstvy *Lesy* (vrstvy připraveny v okně *Table Of Contents*). Název výstupní vrstvy pojmenovat *Lesy_prijmeni*. Na závěr diagram spustit.



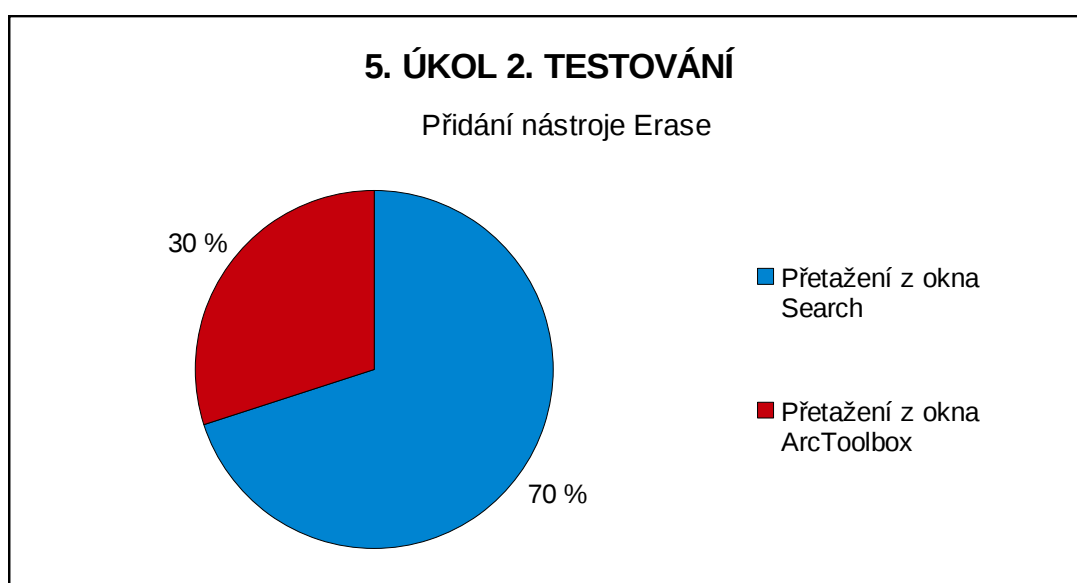
Obr. 6.15 Splněný úkol č. 5 druhého testování interaktivní práce (ModelBuilder).

Nástroj *Erase* v podstatě vykoná opačnou operaci než nástroj *Clip*. Sestavení diagramu s využitím obou nástrojů je obdobné (viz Obr. 6.7 a 6.15). Lze tedy tento úkol porovnat se 4. úkolem prvního testování, kde měli respondenti vytvářet diagram s nástrojem *Clip*. Respondenti dosáhli v tomto úkole 100 % úspěšnosti. Průměrný čas vypracování úkolu byl dvě minuty a 21 sekund. Úkol č. 4 prvního testování měl téměř shodný průměrný čas. Respondentům trvalo průměrně vypracovat úkol jen o jednu sekundu rychleji, ale jejich úspěšnost byla pouze 50 %.

Všichni respondenti správně pojmenovali nově vytvořený model a až na jednoho testovaného postupovali stejné. Zavřeli model a následně provedli pomocí volby *Rename* přejmenování v okně *ArcToolbox*. Dva respondenti pojmenovávali vytvořený model až po vypracování úkolu (Příloha 1, Obr. 12). Respondent P11 postupoval při vytvoření modelu zcela odlišně. Z nástrojové lišty v hlavním okně *ArcMap* vybral tlačítko pro otevření okna *ModelBuilder*, v něm následně vytvořil zadaný diagram. Po vypracování model uložil pomocí volby *Save* z nástrojové lišty okna *ModelBuilder*. Nevýhodou tohoto postupu bylo, že respondent musel v dialogovém okně *Save* zdlouhavě hledat příslušný toolbox, kde se měl vytvořený model uložit. Naopak výhodou byla eliminace nepřejmenování nově vytvořeného modelu.

U 70 % respondentů bylo dalším krokem najít nástroj *Erase* (Příloha 2, Tab. 10). Pro jeho hledání bylo většinou použito okno *Search* (80 %). Přetáhnout nástroj z tohoto okna dokázalo sedm respondentů (viz Obr. 6.16). Jeden respondent sice využil k hledání okno *Search*, ale nevěděl, že z něj jde nástroj rovnou přetáhnout do okna *ModelBuilder*.

Většina respondentů (70 %) přetáhla příslušné vrstvy z okna *Table Of Contents*. Pět respondentů využilo ke spojení prvků spojovací šipky (nástroj *Connect* z nástrojové lišty). Pět respondentů provedlo propojení pomocí dialogového okna nástroje. Učinili tak i dva respondenti, kteří již měli přetažené vrstvy z okna *Table Of Contents*, ale vybrali správné vrstvy (s modrým symbolem), proto se jim nevytvořily duplicitní vrstvy v okně *ModelBuilder*. Dva respondenti se snažili přetáhnout vrstvy z okna *Table Of Contents* přímo do dialogového okna nástroje, to ale není možné.



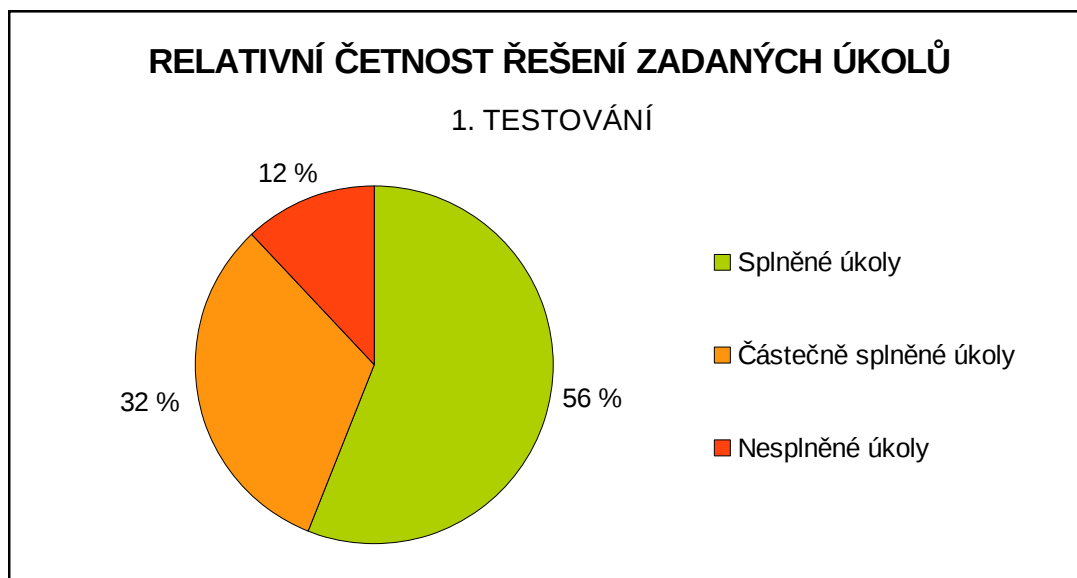
Obr. 6.16 Přidání nástroje *Erase* v 5. úkolu druhého testování interaktivní práce (10 respondentů).

6.3.3 Shrnutí

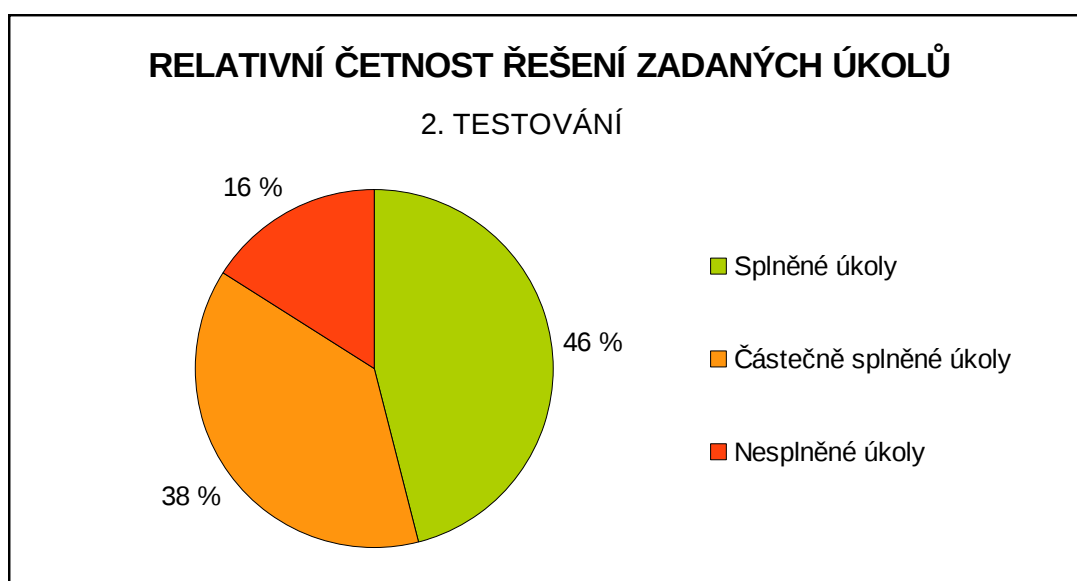
Testování interaktivní práce ve VPL komponentě *ModelBuilder* ukazuje, že studenti 1. ročníku navazující magisterského oboru Geoinformatika (první testování, viz Obr. 6.17) měli o 10 % větší úspěšnost v plnění úkolů než studenti 2. ročníku bakalářského studia Geoinformatika a geografie (druhé testování, viz Obr. 6.18). Nejde, ale jednoznačně určit, že jejich znalosti jsou vyšší než u respondentů z druhého testování, což ani nebylo cílem testování.

Vyšší úspěšnost respondentů v prvním testování může být dána tím, že obsahovalo jednodušší úkoly. Snahou bylo vytvořit dva testy s odlišnými, ale podobně obtížnými úkoly. Pro druhé testování již bylo problematictější vytvořit jednoduché úkoly, poněvadž byly převážně vyčerpány v prvním testování. Do druhého testování byl

zařazen úkol s iterátorem, který byl ze všech úloh nejobtížnější. Tento úkol nezvládlo 60 % respondentů (Příloha 1, Obr. 10).



Obr. 6.17 Relativní četnost řešení zadaných úkolů prvního testování interaktivní práce (10 respondentů).

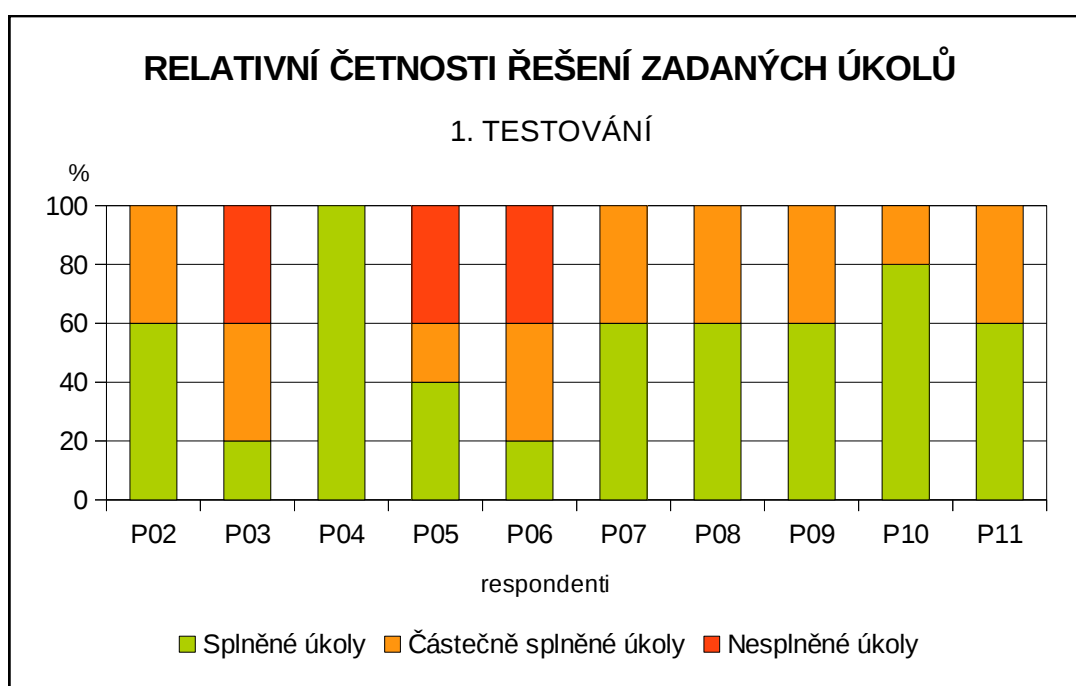


Obr. 6.18 Relativní četnost řešení zadaných úkolů druhého testování interaktivní práce (10 respondentů).

Na druhou stranu lze konstatovat, že v obou testováních neúspěšnost jednotlivých respondentů nepřesáhla 20 % (Obr. 6.19, Obr. 6.20). Znalosti této VPL komponenty studentů na Katedře geoinformatiky jsou dobré, ale bylo zjištěno i pár nedostatků. Zpravidla platí, že respondenti mají pro jednotlivé úkony zažité své postupy, podle kterých pracují. I když je více možných způsobů řešení, které jsou někdy i výhodnější, respondenti přesto preferují jeden a zbylé obvykle ani neznají. Z toho plyne, že pokud

respondent udělá nějakou chybu a nastane podobná situace, chyba je respondentem opakována.

Při vyhodnocování interaktivní práce bylo zjištěno, že značný problém ve VPL komponentě je pojmenování výstupů. Respondenti v mnoha případech neměli přehled o tom jestli pouze přejmenovávají prvek v diagramu či přímo výstup, který se vytvoří po spuštění diagramu. Obdobné problémy měli respondenti i s přejmenováním nově vytvořených modelů. Toto poukazuje na nejednoznačnost ze strany prostředí aplikace *ModelBuilder*. Dalo by se uvažovat o úpravě aplikace tak, aby se uživatelé dopouštěli méně chyb. U přejmenování modelu byl tento problém řešen ve formě přidání volby *Save As*. Tento způsob ukládání modelu je dostupný v aplikaci ArcMap teprve od verze 10.1. Z výsledků testování nelze jednoznačně určit, zda studenti pouze nemají tuto volbu zatím dobře osvojenou nebo je tato úprava funkčnosti aplikace nedostačující.

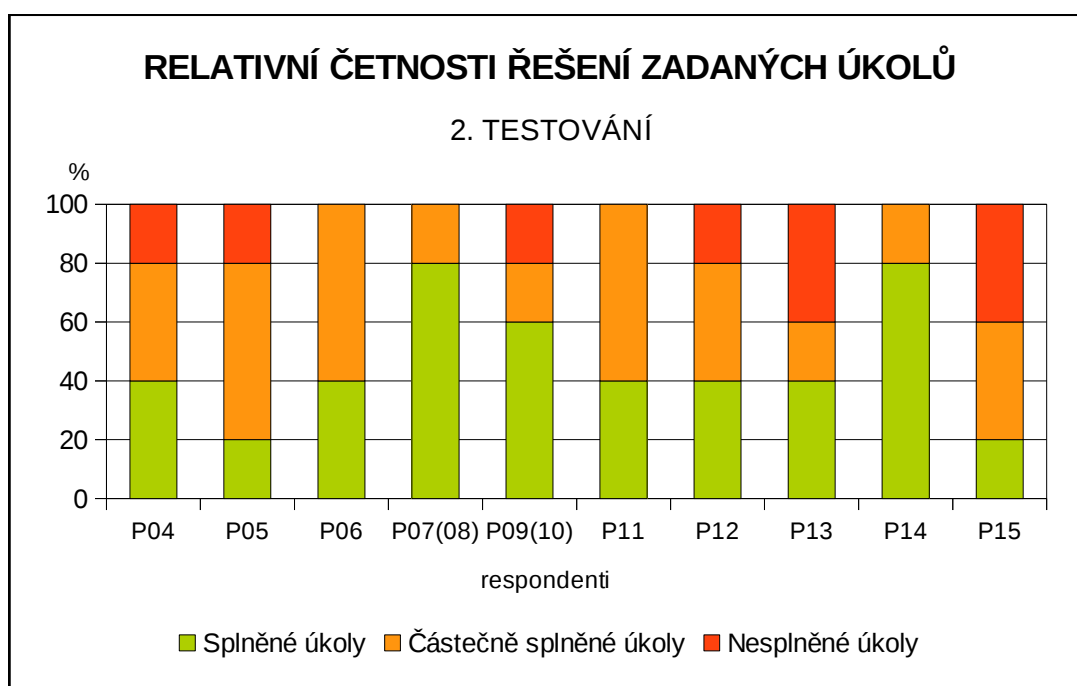


Obr. 6.19 Relativní četnosti řešení zadaných úkolů jednotlivých respondentů v prvního testování interaktivní práce (5 úkolů).

Také bylo vyzorováno, že respondenti nejsou v některých případech schopni plně využít funkcionalitu aplikace. Většinou nastal problém při hledání méně známého nástroje. Někteří respondenti příliš dlouho hledali nástroj v okně *ArcToolbox* než jim došlo, že mohou využít pro vyhledávání i okno *Search*. Následně ale některým chyběla znalost, že lze z okna *Search* nalezený nástroj přímo použít případně přetáhnout rovnou do diagramu. Vrátili se tedy do okna *ArcToolbox* a opět následovalo zdlouhavé hledání nástroje.

Zajímavé bylo pozorovat, jak respondenti pracovali s vrstvami připravenými v okně *Table Of Contents* a jaký způsob následně zvolili pro propojení jednotlivých

prvků diagramu. Většina respondentů vrstvy přímo přetahovala do okna *ModelBuilder* a následně spojovala pomocí spojovacích šipek (*Connect*). Někteří respondenti si vrstvy přetáhli, ale následně se rozhodli využít na propojení prvků dialogové okno nástroje. Ve většině případů zde nastal problém a respondentům se vytvořili v okně *ModelBuilder* duplicitní prvky. Důvodem byla při výběru v dialogovém okně neznalost symbolů u jednotlivých vrstev. Podle toho jestli již byly nebo nebyly přetaženy vrstvy do okna *ModelBuilder*, bylo potřeba brát ohled na výběr vrstev buď s modrým (vrstvy v okně *ModelBuilder*) nebo žlutým symbolem (vrstvy v okně *Table Of Contents*). Pokud měl respondent již přetažené vrstvy v okně *ModelBuilder* a chtěl pro spojení prvků v diagramu využít dialogové okno nástroje, bylo v něm potřeba vybrat vrstvy s modrým symbolem. Jedině tak mohlo být zaručeno, že nebudou vytvořeny duplicitní vrstvy v okně *ModelBuilder*, které by se musely následně smazat.



Obr. 6.20 Relativní četnosti řešení zadaných úkolů jednotlivých respondentů v druhém testování interaktivní práce (5 úkolů).

V prvním testování interaktivní práce byl jeden úkol zaměřen na tlačítko *Auto Layout*, které se používá pro automatické zarovnání diagramů (viz Příloha 1, Obr. 3). Někteří studenti mají tuto funkci velmi dobře osvojenou a využili ji také při řešení dalších úloh, obdobně ji používali i respondenti druhého testování. Bylo ale s podivem, že tlačítko *Auto Layout* někteří respondenti nevyužili. U nich nastaly problémy zejména v případech, kdy se jim přidal nový prvek do diagramu přes již stávající. Když se nově přidaný prvek snažili přemístit jinde, byl zároveň vybrán i prvek pod ním, bylo tedy pohybováno oběma prvky najednou, což bylo nežádoucí. Zde by jednoznačně bylo výhodné použít tlačítko *Auto Layout*.

7 VÝSLEDKY

Diplomová práce se zabývá eye-tracking testováním vizuálních programovacích jazyků. Práce obsahuje dvě hlavní části, první se zabývá hodnocením vizuální salience a druhá testováním interaktivní práce.

Prvním krokem bylo vytvoření diagramů datových toků ve vybraných vizuálních programovacích jazycích pro testování vizuální salience. Pro tuto část byly zvoleny čtyři VPL komponenty: Model Maker, Macro Modeler, ModelBuilder a Workflow Designer. V každé VPL komponentě bylo vytvořeno 12 diagramů (Příloha 4). Za účelem testování salience metodou eye-tracking bylo celkem připraveno pět experimentů. Pro respondenty se zkušenostmi s VPL byly vytvořeny experimenty pro každou VPL komponentu zvlášť. Byla naměřena data od 18 respondentů pro Model Maker, 19 respondentů pro Macro Modeler, 21 respondentů pro ModelBuilder a 19 respondentů pro Workflow Designer. Nakonec byl vytvořen jeden experiment pro respondenty bez zkušeností s vizuálními programovacími jazyky obsahující experimentální stimuly všech VPL komponent, získána byla data od 21 respondentů.

Dále proběhlo testování Saliency modulu, který je součástí open source programu OGAMA a vytvořeny ukázky saliency map (Příloha 3). Následně byla vypočítána vizuální salience pro všech 48 experimentálních stimulů pomocí programu OGAMA (Příloha 6). Tyto výstupy byly porovnány s výsledky testování vizuální salience metodou eye-tracking. Z každé VPL komponenty byly vybrány do textové části dva experimentální stimuly, které byly vhodně vizualizovány a slovně okomentovány. Bylo zjištěno, že testování salience metodou eye-tracking je do velké míry ovlivňováno způsobem čtením diagramů. Z toho důvodu se vypočítaná salience a naměřená salience metodou eye-tracking zpravidla neshodují. Ale i přesto byla shoda v některých případech zjištěna, zejména u VPL komponenty ModelBuilder. Nakonec byly srovnány výsledky eye-tracking testování podle zkušeností s VPL a bylo zjištěno, že obě skupiny respondentů postupují při čtení diagramů stejným způsobem. Na závěr byl o této části diplomové práce vytvořen poster (Příloha 5).

Druhým hlavním cílem práce bylo testování a vyhodnocení interaktivní práce ve vizuálních programovacích jazycích. Pro tuto část práce byla vybrána jedna VPL komponenta ModelBuilder. Následně byly sestaveny dva experimenty, pro které byly nachystány sady po pěti úkolech. Pro vybrané úlohy bylo potřeba předem připravit diagramy (Příloha 4). Prvního testování se celkem zúčastnilo deset respondentů a druhého 12 respondentů. Během vyhodnocování interaktivní práce byly vytvořeny přehledové tabulky jednotlivých úkonů všech respondentů (Příloha 7 a Příloha 8) a sestaveny tabulky časů pro zvolené úkony prováděné v jednotlivých úkolech (Příloha 2). Pro vybrané operace byly vytvořeny grafy (Příloha 1). Všechny úkoly byly následně opatřeny slovním komentářem.

8 DISKUZE

Cíle této práce lze rozdělit na dvě velké hlavní části. První bylo eye-tracking testování a srovnání vizuální salience ve vizuálních programovacích jazycích. Další částí bylo vyhodnocení interaktivní práci ve vybrané VPL komponentě.

Pro vyhodnocení salience byly vybrány čtyři VPL komponenty. V uvažování bylo rozšířit výběru o další VPL komponentu Spatial Model Editor, která je nově součástí programu ERDAS Imagine 2013 a již byla zahrnuta do výuky na Katedře geoinformatiky. Z ohledem na rozsah práce a časové možnosti nebyla tato VPL komponenta do testování zahrnuta. Místo toho byla testována VPL komponenta Model Maker, která jako jediná obsahuje v diagramech pouze černobílé prvky, což bylo pro hodnocení přínosem.

Důležitou částí práce bylo seznámení s programem OGAMA a samotnou metodou Saliency. Tento krok byl podstatný pro samotnou tvorbu salience map. Pro program, ale není dostupný dostatečně podrobný návod, který by popisoval jednotlivé parametry nastavení. Návod Saliency modulu odkazuje na webové stránky tvůrce této metody, kde jsou dostupné přímo zdrojové kódy. Bohužel, ale nebylo možné zjistit, které části jsou přesně implementovány do Saliency modulu. S ohledem na dvě velké témata, které tato práce pojímá, nebylo v možnostech autorky práce zdrojové kódy podrobněji prostudovat, což by, ale mohlo být přínosné.

Výsledky generované salience (saliency mapy) a výsledky testování vizuální salience metodou eye-tracking (heatmapy) byly porovnávány pouze vizuálně. Snahou bylo jednotlivé výstupy propojit s využitím aplikace ArcMap pomocí nástroje *Raster Calculator*. Výsledkem byl, ale neinterpretovatelný výstup. Další pokusy o spojení výsledků probíhalo v grafickém editoru GIMP. Od obou těchto variant bylo pro jejich nevhodnost nakonec upuštěno.

Bylo uvažováno i nad dalšími možnostmi, jak vizualizovat výsledky eye-tracking testování. V úvahu připadala metoda scanpath, porovnání scanpathů jednotlivých respondentů s predikovaným scanpathem vypočítaným pomocí metody Saliency v programu OGAMA. V tomto směru by mohla být práce dále rozvíjena.

Druhou hlavní částí práce bylo hodnocení interaktivní práce ve vybrané VPL komponentě. Obtížné bylo již samotné sestavování úkolů. Záměrem bylo sestavit jednoduché a ne příliš náročné úkoly. Následně bylo potřeba vytvořit zadání jednotlivých úkolů tak, aby byli pro respondenty jasné a výstižné. U sestavování druhého testování interaktivity bylo již velmi složité vytvořit úkoly tak, aby se neopakovaly ty z prvního testování a zároveň byly dostatečně jednoduché. Čím složitější byl totiž úkol, tím bylo problematictější jednoduše a stručně sestavit zadání, tak, aby nezahltilo respondenta.

Interaktivní testování metodou eye-tracking je obtížné pro operátora nejen při vyhodnocování, ale již při samotném průběhu testu. Je potřeba neustále sledovat, v které části testu se respondent právě nachází a podle toho maximalizovat jednotlivá předem minimalizovaná okna aplikací. Přes všechnu snahu se nepodařilo zachovat stejné výchozí podmínky pro všechny respondenty, z tohoto důvodu byli vyrazeni z hodnocení druhého testování interaktivní práce dva respondenti. U druhého testování interaktivní práce vyvstaly také technické problémy, kdy došlo k neočekávanému ukončení celé aplikace ArcMap, v které právě probíhalo testování. Problémy pokračovaly u dalšího respondenta ve formě hlášení o chybě skriptu. Testování muselo být v obou těchto případech rozděleno na dvě části.

Vyhodnocování získaných dat je časově velmi náročné. Bylo potřeba shlédnout videa jednotlivých úkolů od všech respondentů a zaznamenat významné činnosti a jejich časy. Z tohoto důvodu se práce nevěnuje dalším možnostem vyhodnocení, což by mohly být například oblasti zájmu (AOI).

Nevýhodou na samotném testování interaktivní práce bylo, že respondenti nedostali následně zpětnou vazbu o správnosti řešení zadaných úkolů. Někteří respondenti se dožadovali po vypracování úkolu správného řešení. Testování by se tak stalo prospěšné i pro samotné studenty, zjistili by, v čem dělají chyby a mají nedostatky. Zaznamenané videa z testování interaktivní práce by mohly být dobrou učební pomůckou pro zlepšení dovedností v daném programu. Při sledování videa lze zjistit mnoho nových poznatků. Při opakovaném sledování stejného úkolu, ale řešeného různými respondenty je možno získat výbornou znalost o nejlepšímu postupu, jak úlohu správně vypracovat. Tímto způsobem by šlo získat dobré znalosti o daném programu, protože chybami se člověk učí a co by viděl pozorovatel, že respondentovi nejde, mohl by si sám zkusit v programu vyřešit nebo shlédnout záznam od dalšího respondenta, který by problém třeba vyřešil.

Na závěr by bylo dobré sdělit, že tato diplomová práce obsahuje na sobě dvě nezávislá témata, které spojuje testování metodou eye-tracking a vizuální programovací jazyky. Z toho důvodu bylo velmi obtížné v rozsahu práce obsáhnout všechny důležité poznatky, jak z oblasti vizuální salience, tak oblasti testování interaktivní práce.

Diplomová práce může být přínosná množstvím naměřených dat z testování vizuální salience či interaktivní práce metodou eye-tracking. Data mohou být dále analyzována v rámci dalších prací.

9 ZÁVĚR

Cílem diplomové práce bylo zhodnotit vizuální salienční a interaktivní práci ve vizuálních programovacích jazycích používaných v GIS pomocí metody eye-tracking. Hlavní náplní práce byla příprava, průběh a vyhodnocení jednotlivých eye-tracking testů. Dále seznámení s open source programem OGAMA a metodou Saliency.

Pro praktickou část práce zabývající se hodnocením salience ve vizuálních programovacích jazycích byly vybrány čtyři aplikace obsahující VPL komponenty: Model Maker (ERDAS Imagine 2013), Macro Modeler (Idrisi 17.0 The Selva Edition), ModelBuilder (ArcGIS 10.1 for Desktop Advanced) a Workflow Designer (Autodesk AutoCAD Map 3D 2014). Programy byly vybrány s ohledem na programové vybavení Katedry geoinformatiky Univerzity Palackého v Olomouci.

V rámci práce bylo vytvořeno a realizováno pět testů pro hodnocení vizuální salience. Experimentální stimuly byly ve formě obrázků, které představovaly diagramy datových toků jednotlivých VPL komponent. Testování salience metodou eye-tracking se účastnili studenti studující na Katedře geoinformatiky. Cílem bylo otestovat především respondenty, kteří mají zkušenosti s vizuálními programovacími jazyky. Z tohoto důvodu bylo potřeba nejdříve seznámit studenty s VPL komponentami Model Maker a Macro Modeler, které se na Katedře geoinformatiky běžně nevyučují.

Po otestování Saliency modulu, který je součástí programu OGAMA, byly vypočítány salienční mapy pro jednotlivé experimentální stimuly všech VPL komponent. Výsledky byly následně porovnány s naměřenými daty z testování salience metodou eye-tracking. Posledním krokem této části práce bylo srovnání dat z eye-tracking testování podle zkušeností s VPL komponentami. Po vyhodnocení a srovnání salience bylo zjištěno, že respondenti byli při testování ve velké míře ovlivněni způsobem čtení diagramů nikoliv samotnou nápadností prvků.

Druhou hlavní částí práce bylo vyhodnocení interaktivní práce ve vybrané VPL komponentě ModelBuilder. Celkem byly připraveny a realizovány dva interaktivní testy s experimentálními stimuly typu snímání obrazovky. Následně probíhalo vyhodnocování získaných dat z testování interaktivní práce metodou eye-tracking. Zjišťovány byly jednotlivé činnosti a časy provedení těchto úkonů a to u všech testovaných a všech zadaných úkolů. Zjištěné informace byly zaznamenány do přehledových tabulek. Vybrané úkony byly následně zobrazeny v grafech. Jednotlivé úkoly byly slovně zhodnoceny.

Na závěr byly vytvořeny webové stránky o této diplomové práci. Ty jsou umístěny na serveru Katedry geoinformatiky.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

Použitá literatura

DUCHOWSKI, Andrew, 2007. *Eye tracking methodology: theory and practice*. Second Edition. London: Springer, xxiv, 328 s. ISBN 978-1-84628-608-7.

HOLMQVIST, Kenneth, Marcus NYSTRÖM, Richard ANDERSSON, Richard DEWHURST, Halszka JARODZKA a Joost VAN DE WEIJER, 2011. *Eye tracking: a comprehensive guide to methods and measures*. New York: Oxford University Press, xix, 537 s. ISBN 978-0-19-969708-3.

MARRIOTT, Kim a Bernd MEYER, 1998. *Visual language theory*. New York: Springer, xi, 381 s. ISBN 03-879-8367-8.

Informační zdroje

BORJI, Ali, Dicky N. SIHITE a Laurent ITTI. Quantitative analysis of human-model agreement in visual saliency modeling: a comparative study. In: *IEEE transactions on image processing* [online]. 2013, roč. 22, č. 1, s. 55-69 [cit. 2014-04-11]. Dostupné z: http://ilab.usc.edu/publications/doc/Borji_etal12tip.pdf

DOBEŠOVÁ, Zdena. Visual programming language in geographic information systems. *Recent Researches in Applied Informatics* [online]. 2011, s. 276-280 [cit. 2014-04-11]. ISBN 978-1-61804-034-3. Dostupné z: <http://www.wseas.us/library/conferences/2011/Prague/AICT/AICT-47.pdf>

DOBEŠOVÁ, Zdena a Petr DOBEŠ. Differences in visual programming for GIS. *Applied Mechanics and Materials*. 2014, 519-520, s. 355-358. DOI: 10.4028/www.scientific.net/AMM.519-520.355.

Eye-tracking at Department of Geoinformatics Palacký University in Olomouc, Czech Republic [online]. © 2012 [cit. 2014-04-11]. Dostupné z: <http://eyetracking.upol.cz/>

ILab Neuromorphic Vision C++ Toolkit [online]. © 2002-2004, Last updated Thursday, 01-Aug-2013 [cit. 2014-04-11]. Dostupné z: <http://ilab.usc.edu/toolkit/>

ITTI, Laurent. Visual salience. In: *Scholarpedia: the peer-reviewed open-access encyclopedia* [online]. 2007-09-07, last modified on 16 January 2010 [cit. 2013-06-10]. Dostupné z: http://www.scholarpedia.org/article/Visual_salience

ITTI, Laurent, Christof KOCH a Ernst NIEBUR. A model of saliency-based visual attention for rapid scene analysis. In: *IEEE transactions on pattern analysis and machine intelligence* [online]. Los Alamitos: IEEE Computer Society, 1998, roč. 20, č. 11, s. 1254-1259 [cit. 2014-04-11]. ISSN 0162-8828. Dostupné z: http://ilab.usc.edu/publications/doc/Itti_etal98pami.pdf

KOHOUTEK, Rudolf. Slovník věd o člověku - S. In: *Psychologie v teorii a praxi* [online]. 8. prosince 2008 [cit. 2014-04-11]. Dostupné z: <http://rudolfkohoutek.blog.cz/0812/sloivnik-ved-o-cloveku-s>

KOTYZ, Pavel. *Implementace nástrojů SMI Experiment Suite 360° pro hodnocení mapové kompozice* [online]. Olomouc, 2013 [cit. 2014-04-10]. Dostupné z: <http://www.geoinformatics.upol.cz/dprace/bakalarske/kotyz13/>. Bakalářská práce. Univerzita Palackého v Olomouci.

KUDEĽKA, Václav. *Srovnání vizuálního programování v GIS produktech podle kognitivních dimenzí* [online]. Olomouc, 2013 [cit. 2013-06-10]. Dostupné z: <http://www.geoinformatics.upol.cz/dprace/bakalarske/kudelka13/>. Bakalářská práce. Univerzita Palackého v Olomouci, Přírodovědecká fakulta, Katedra geoinformatiky.

LUKAUSKÝ, Jiří. *Pozornost* [online]. Praha: Psychologický ústav AV ČR, 2009 [cit. 2014-04-11]. Dostupné z: https://is.muni.cz/el/1421/podzim2009/PSA_005/um/01-Brno-pozornost-2009.pdf

NAVALPAKKAM, Vidhya a Laurent ITTI. Modeling the influence of task on attention. In: *Vision Research* [online]. 2005, roč. 45, č. 2, s. 205-231 [cit. 2014-04-11]. Dostupné z: http://ilab.usc.edu/publications/doc/Navalpakkam_Itti05vr.pdf

NIEBUR, Ernst. Saliency map. In: *Scholarpedia: the peer-reviewed open-access encyclopedia* [online]. 2007-08-28, last modified on 6 June 2010 [cit. 2014-04-11]. Dostupné z: http://www.scholarpedia.org/article/Saliency_map

NIELSEN, Jakob. Why You Only Need to Test with 5 Users. *NN/g Nielsen Norman Group: Evidence-Based User Experience Research, Training, and Consulting* [online]. March 19, 2000 [cit. 2014-03-30]. Dostupné z: <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>

NIELSEN, Jakob. How Many Test Users in a Usability Study?. *NN/g Nielsen Norman Group: Evidence-Based User Experience Research, Training, and Consulting* [online]. June 4, 2012 [cit. 2014-03-30]. Dostupné z: <http://www.nngroup.com/articles/how-many-test-users/>

POPELKA, Stanislav a Alžběta BRYCHTOVÁ. Eye-tracking: sledování (pohybu) očí. [online]. 2013 [cit. 2014-04-11]. Dostupné z:
https://docs.google.com/presentation/d/1eUAKiGIRBJZuGr1w_vanA8ZVsItoYrFl310H9t0xJGI/pub?start=false&loop=false&delayms=60000#slide=id.g256400436_2_9

POPELKA, Stanislav, Alžběta BRYCHTOVÁ a Vít VOŽENÍLEK. Eye-tracking a jeho využití při hodnocení map. *Geografický časopis*. Geografický ústav SAV, 2012, roč. 64, č. 1, s. 71-87. ISSN: 0016-7193. Dostupné z:
<http://www.sav.sk/journals/uploads/07101210Popelka%20et%20al..pdf>

Saliency. In: *Wikipedie: otevřená encyklopedie* [online]. c2013, naposledy editována 11. 3. 2013 [cit. 2014-04-11]. Dostupné z: <http://cs.wikipedia.org/wiki/Saliency>

VOŠKÜHLER, Adrian. *OGAMA Description (for Version 2.5): a software to record, analyze and visualize gaze and mouse movements in screen based environments*. Berlin, 2009 [cit. 2013-03-30]. Dostupné z:
<http://www.ogama.net/sites/default/files/pdf/OGAMA-DescriptionV25.pdf>

VOŠKÜHLER, Adrian. *OGAMA: open gaze and mouse analyzer* [online]. 28.09.2013 [cit. 2014-03-30]. Dostupné z: <http://www.ogama.net/>

WALTHER, Dirk B. *SaliencyToolbox* [online]. 2013 [cit. 2014-04-11]. Dostupné z: <http://www.saliencytoolbox.net/>

ZIEGLEROVÁ, Jana. *Vybrané aspekty uživatelského vnímání kartografických děl* [online]. Olomouc, 2012 [cit. 2013-06-10]. Dostupné z:
<http://geoinformatics.upol.cz/dprace/bakalarske/zieglerova12/>. Bakalářská práce. Univerzita Palackého v Olomouci, Přírodovědecká fakulta, Katedra geoinformatiky.

SUMMARY

The main aim of the master thesis was a eye-tracking evaluation of salience and interactive work in visual programming languages.

The first part of the thesis is about visual salience. The aim was a preparation and realization testing of salience by the method of eye-tracking, and then a comparison of calculated visual salience in software OGAMA and results of eye-tracking testing.

Four visual programming languages for eye-tracking testing were selected. They were Model Maker (ERDAS Imagine 2013), Macro Modeler (Idrisi 17.0 The Selva Edition), ModelBuilder (ArcGIS 10.1 for Desktop Advanced) and Workflow Designer (Autodesk AutoCAD Map 3D 2014). Five experiments for eye-tracking testing of salience were created. Individual test was prepared for four selected VPL components. The number of stimuli was twelve in each experiment except the last one. This experiment contains the data flow diagrams of all four VPL components and the number of stimuli was forty-eight. Last experiment was for respondents with no experience with VPL components. The type of stimuli was image. The images change after two seconds.

Other step was to calculate a visual salience in software OGAMA. The results are saliency maps with the order of fixations. The results of eye-tracking testing of salience and calculated visual salience was compared. The comparison was commented verbally. After evaluation and comparison, it was found that respondents were influenced the way of reading data flow diagrams.

In the second part of the thesis a evaluation of interactive work in selected visual programming language was performed. The testing of interactive work by using method of eye-tracking was held in VPL component ModelBuilder. Two experiments for eye-tracing testing were created. The number of stimuli was five in each experiment. The type of stimuli was screen recording. In first testing the number of participants was ten and in second testing the number of respondents was twelve. The eye-tracking evaluation of interactive work was very time-consuming. It was necessary to watch all records from individual participants. The overview tables of operations were created. For selected operations were written time information into the tables. Then individual tasks were commented verbally.

Finally, a web site of this master thesis was created, which is located at the server the Department of Geoinformatics.

PŘÍLOHY

SEZNAM PŘÍLOH

Vázané přílohy

Příloha 1	Grafy
Příloha 2	Tabulky
Příloha 3	Ukázky saliency map pro předdefinované a vlastní kombinace kanálů

Volné přílohy

Příloha 4	DVD
Příloha 5	Poster

Přílohy na DVD

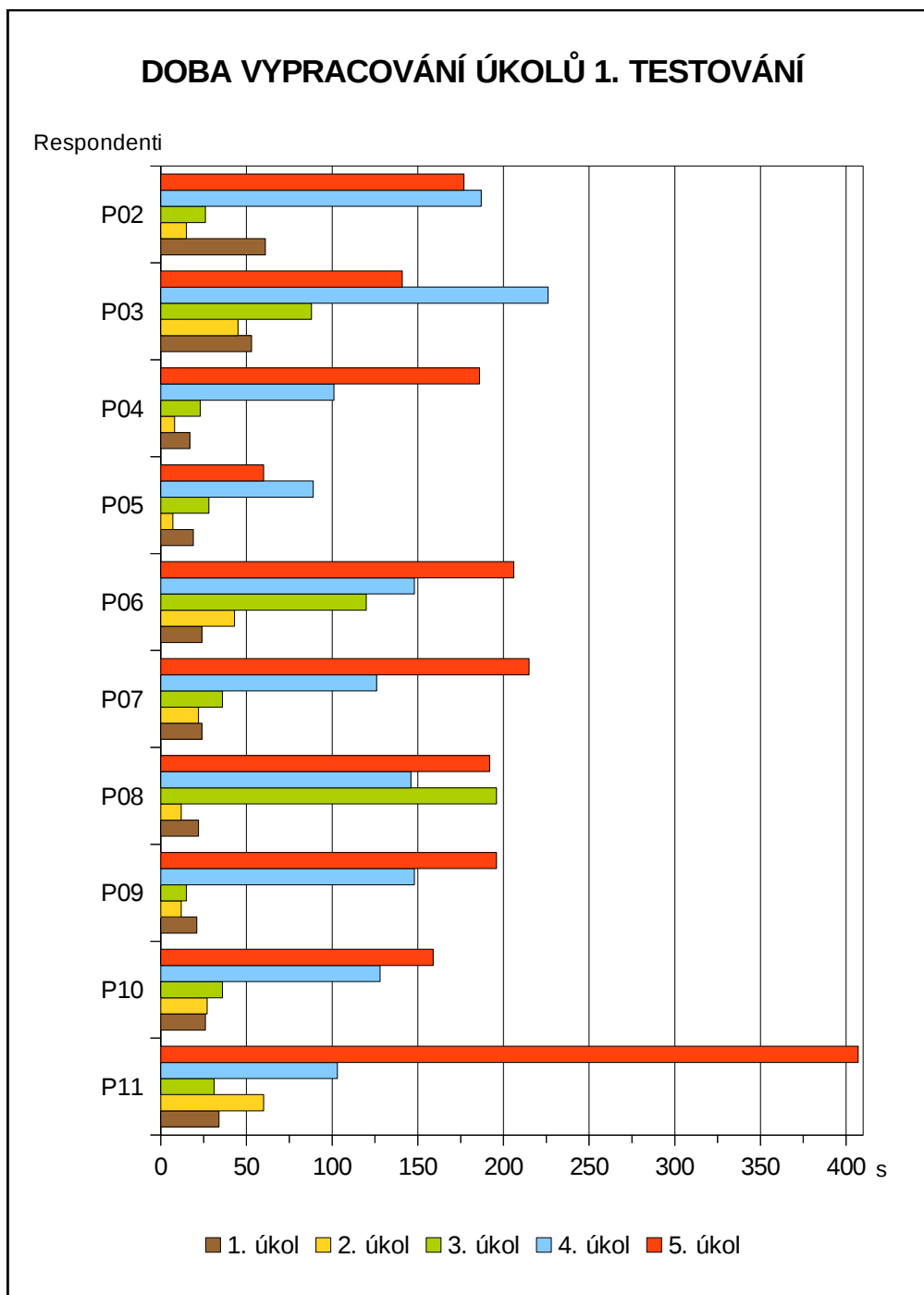
Příloha 6	Saliency mapy diagramů hodnocených VPL komponent
Příloha 7	Přehled úkonů v prvním testování interaktivní práce
Příloha 8	Přehled úkonů v druhém testování interaktivní práce

Popis struktury DVD

Adresáře:

- Diagramy_ArcGIS
- Diagramy_AutoCAD
- Diagramy_Erdas
- Diagramy_Idrisi
- EyeTracking_data
- EyeTracking_projekty_BeGaze
- EyeTracking_projekty_Ogama
- Metadata
- Prilohy
- Text_prace
- Web

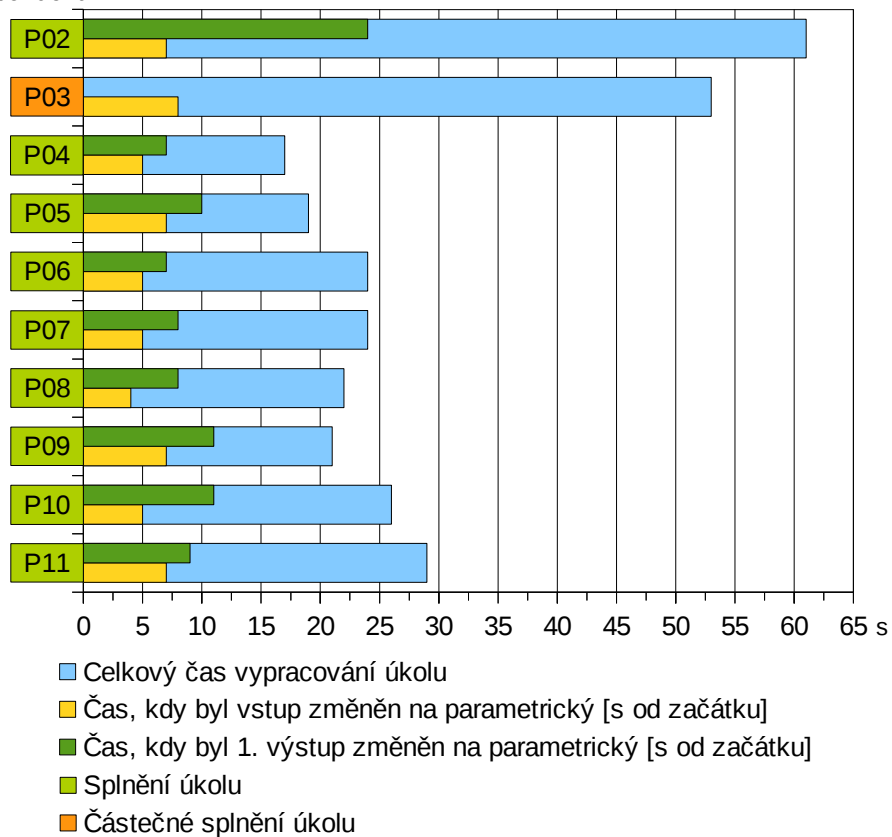
Grafy



Obr. 1 Doba vypracování jednotlivých úkolů prvního testování interaktivní práce metodou eye-tracking.

1. ÚKOL 1. TESTOVÁNÍ

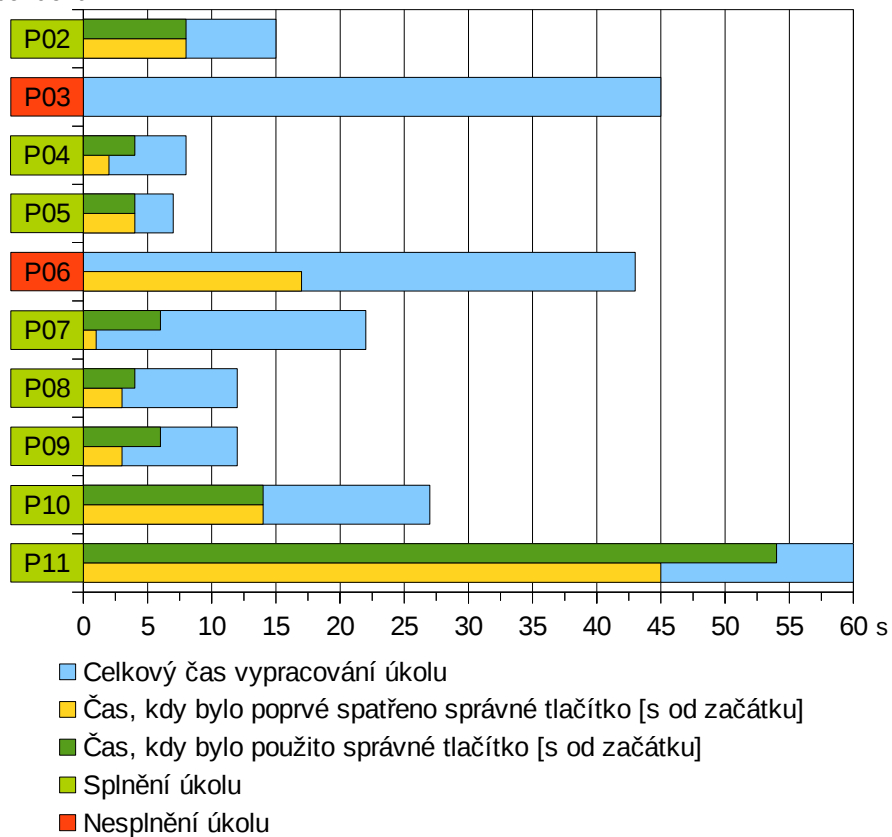
Respondenti



Obr. 2 Úkol č. 1 prvního testování interaktivní práce metodou eye-tracking.

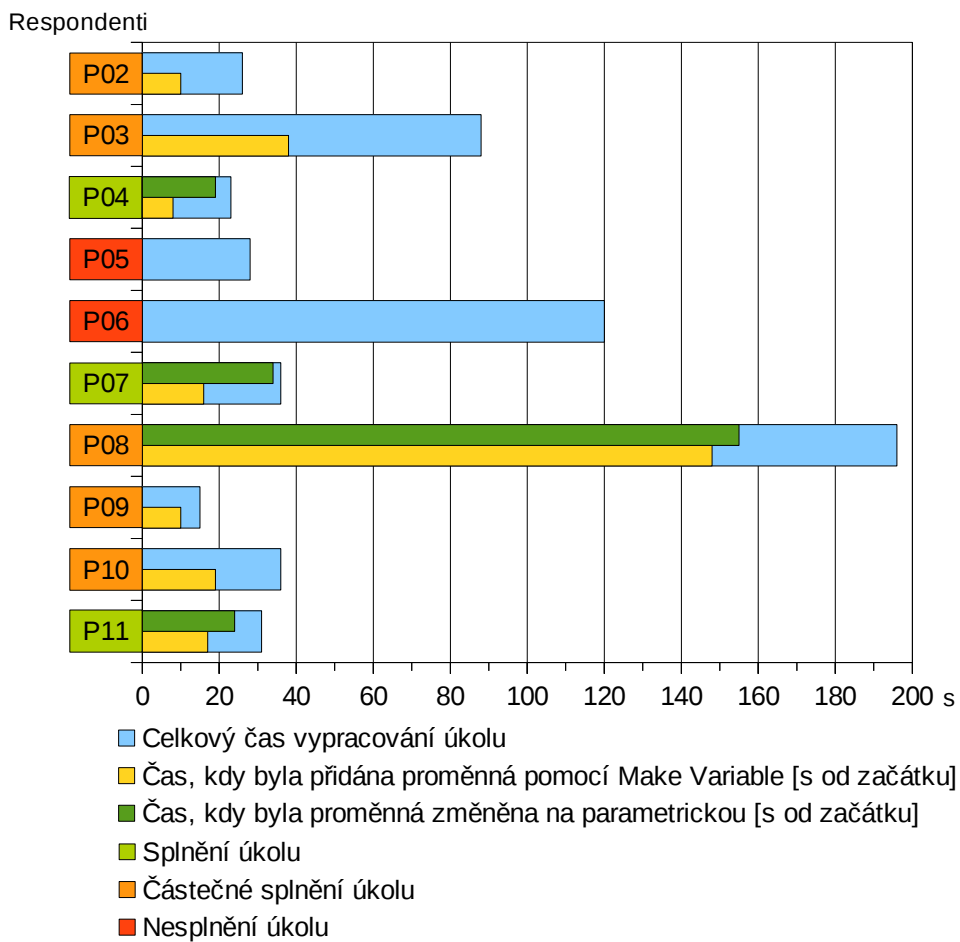
2. ÚKOL 1. TESTOVÁNÍ

Respondenti



Obr. 3 Použití tlačítka *Auto Layout* v 2. úkolu prvního testování interaktivní práce.

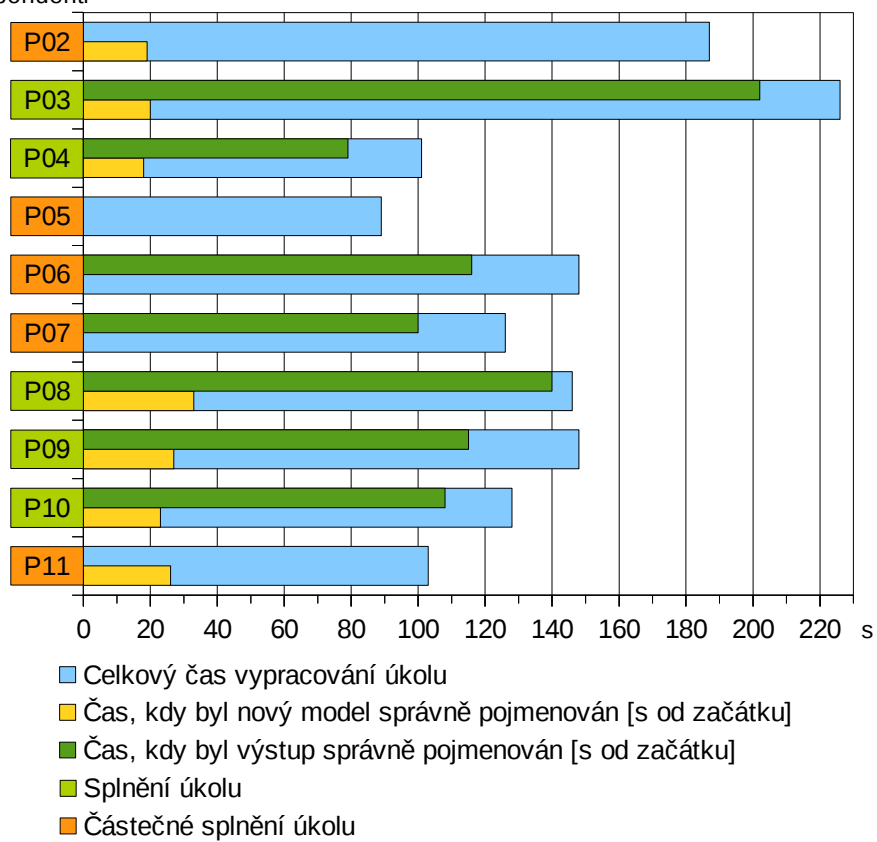
3. ÚKOL 1. TESTOVÁNÍ



Obr. 4 Úkol č. 3 prvního testování interaktivní práce metodou eye-tracking.

4. ÚKOL 1. TESTOVÁNÍ

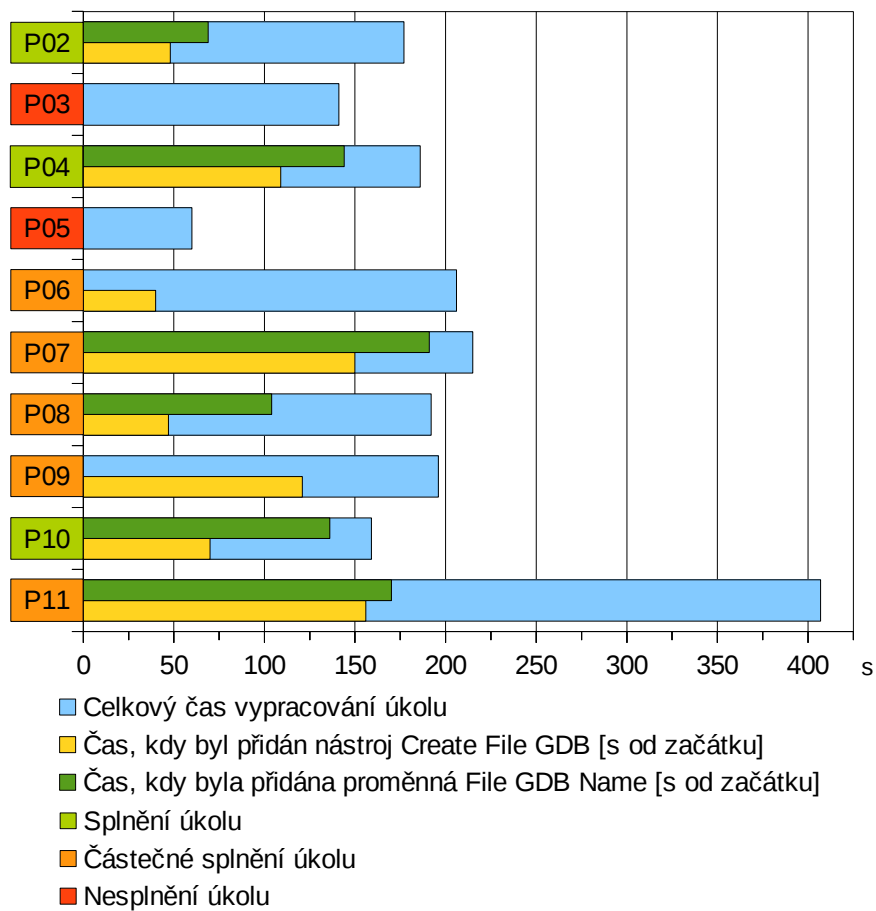
Respondenti



Obr. 5 Úkol č. 4 prvního testování interaktivní práce metodou eye-tracking.

5. ÚKOL 1. TESTOVÁNÍ

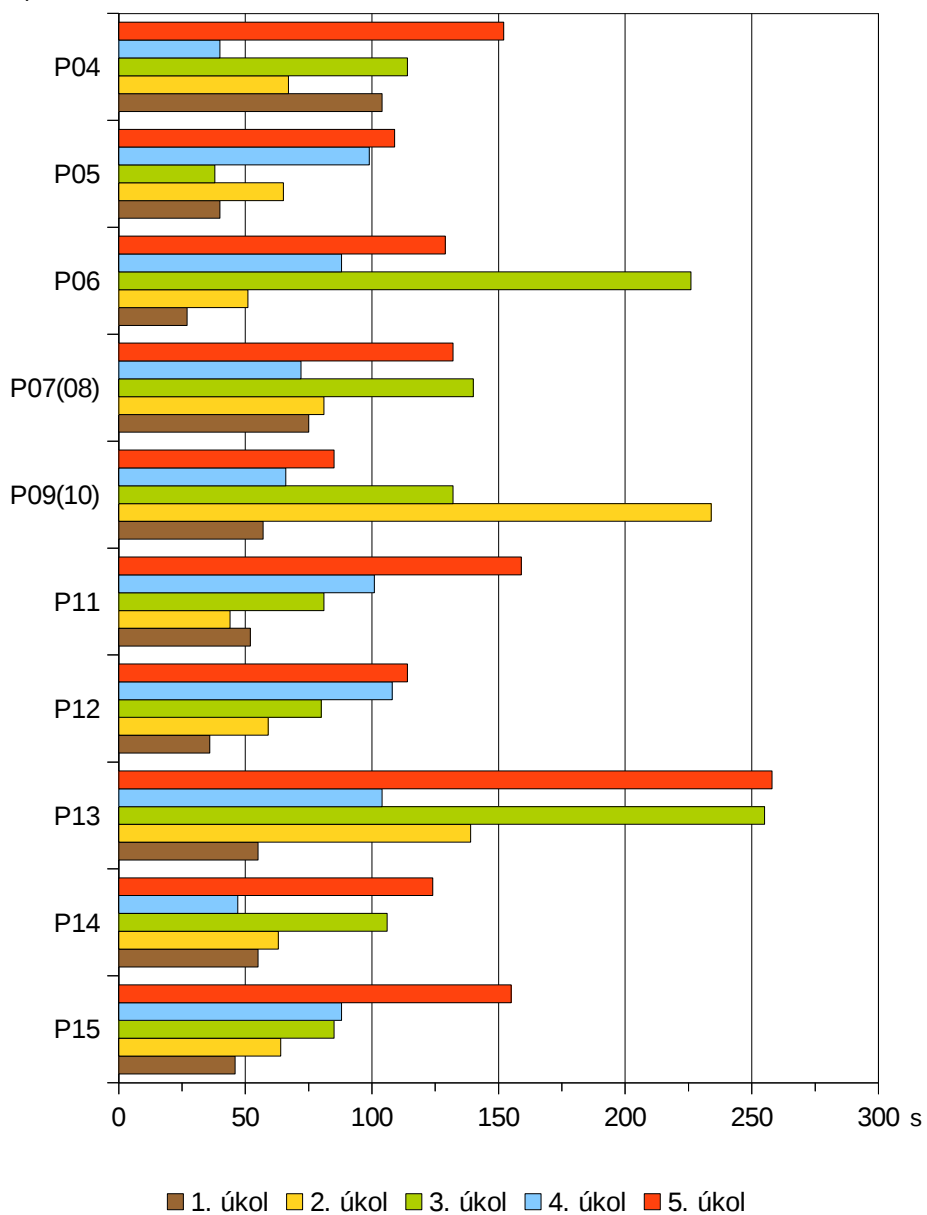
Respondenti



Obr. 6 Úkol č. 5 prvního testování interaktivní práce metodou eye-tracking.

DOBA VYPRACOVÁNÍ ÚKOLŮ 2. TESTOVÁNÍ

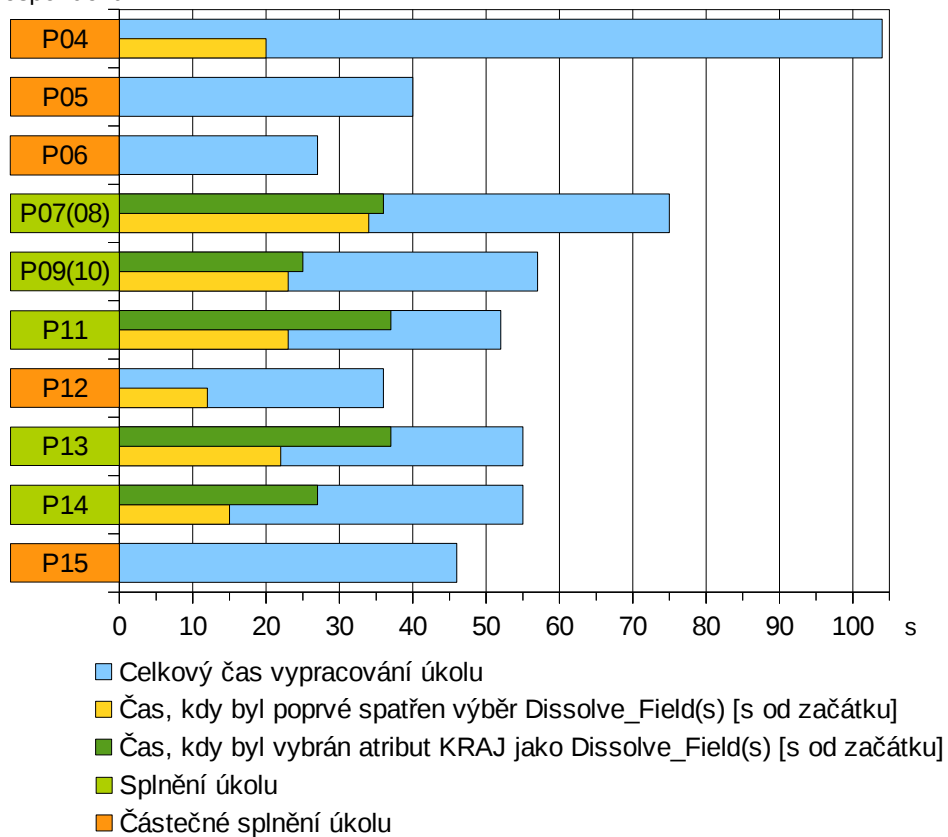
Respondenti



Obr. 7 Doba vypracování jednotlivých úkolů druhého testování interaktivní práce metodou eye-tracking.

1. ÚKOL 2. TESTOVÁNÍ

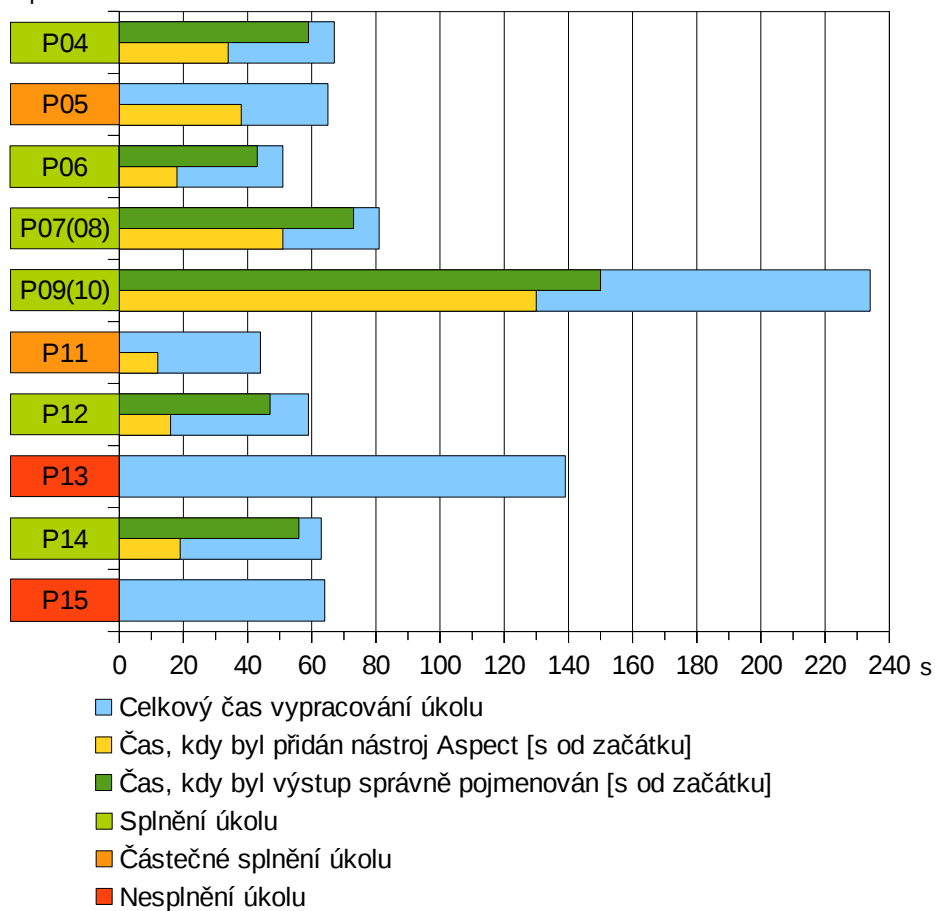
Respondenti



Obr. 8 Úkol č. 1 druhého testování interaktivní práce metodou eye-tracking.

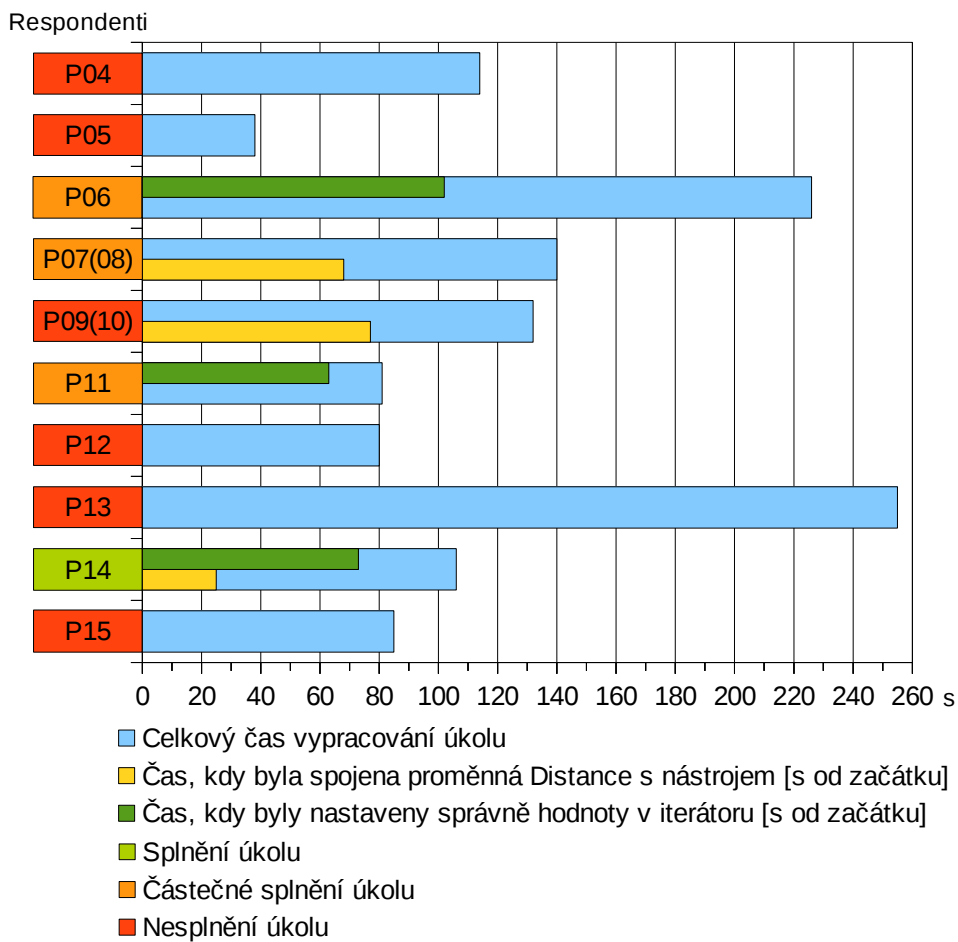
2. ÚKOL 2. TESTOVÁNÍ

Respondenti



Obr. 9 Úkol č. 2 druhého testování interaktivní práce metodou eye-tracking.

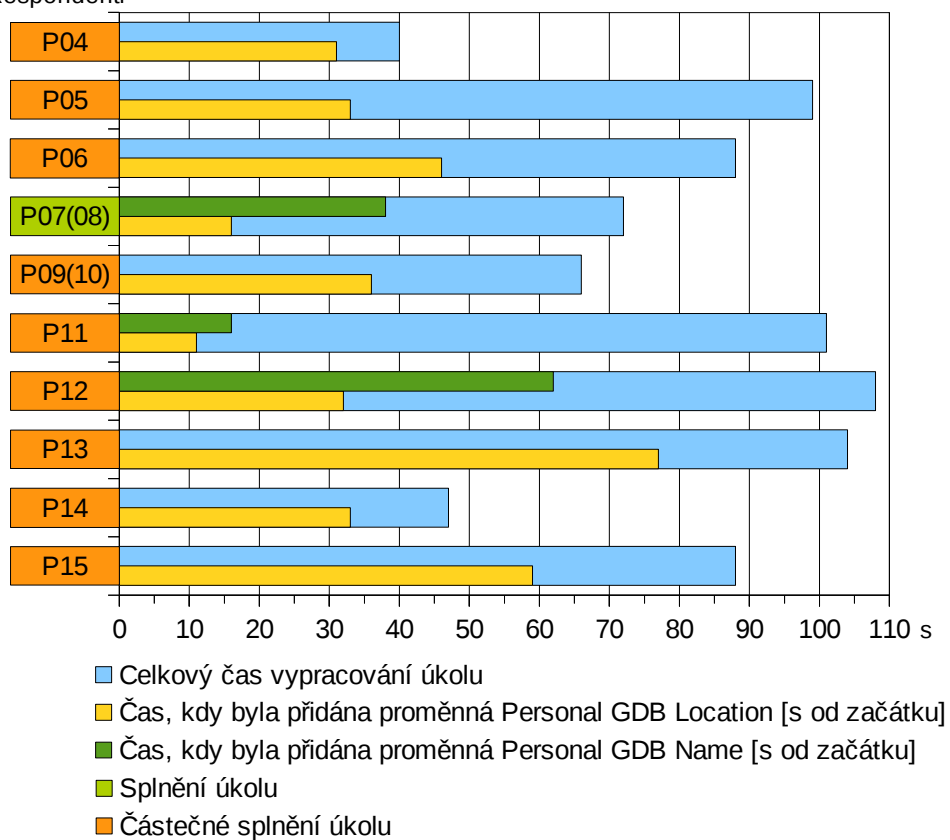
3. ÚKOL 2. TESTOVÁNÍ



Obr. 10 Úkol č. 3 druhého testování interaktivní práce metodou eye-tracking.

4. ÚKOL 2. TESTOVÁNÍ

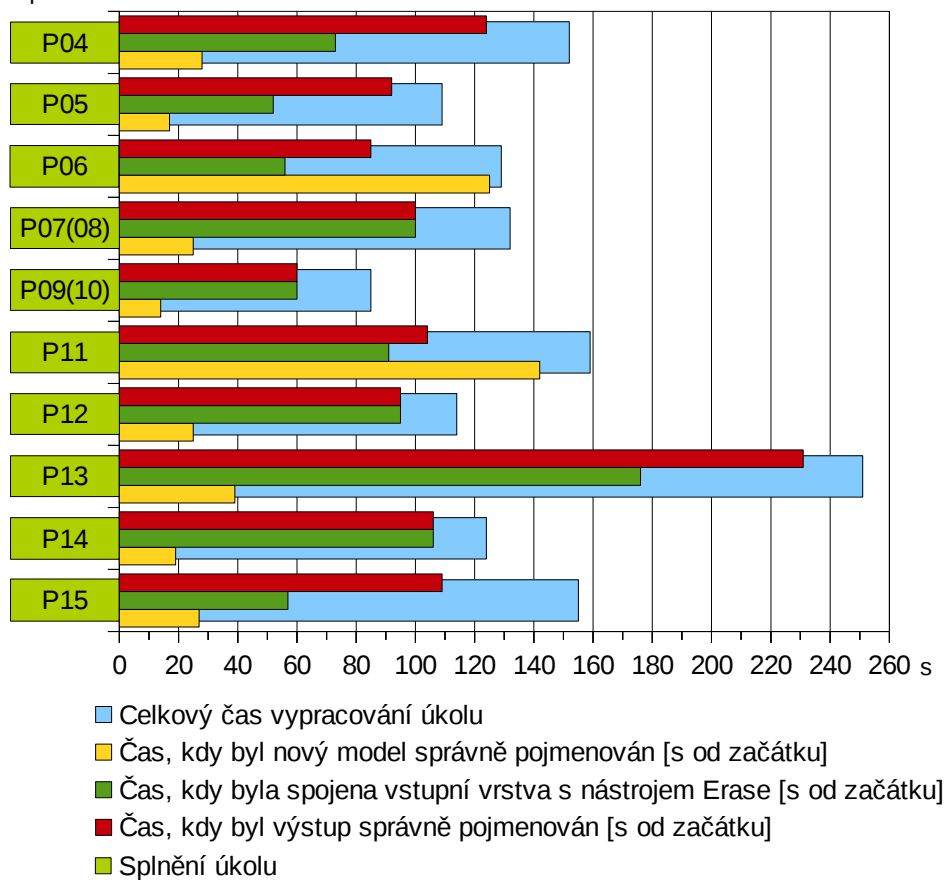
Respondenti



Obr. 11 Úkol č. 4 druhého testování interaktivní práce metodou eye-tracking.

5. ÚKOL 2. TESTOVÁNÍ

Respondenti



Obr. 12 Úkol č. 5 druhého testování interaktivní práce metodou eye-tracking.

Příloha 2

Tabulky

Tab. 1 Zadání úkolů prvního testování interaktivní práce metodou eye-tracking

Úkol	Zadání
1.	<p>V diagramu (<i>01_ukol_dmr</i>) změňte vstupy a výstupy na parametrické vstupy a výstupy.</p> <p>Změny neukládejte.</p> <p>Po splnění úkolu minimalizujte okno <i>ModelBuilder</i> a stiskněte F11.</p>
2.	<p>Prvky v diagramu (<i>02_ukol_auto_layout</i>) automaticky uspořádejte (zarovnejte).</p> <p>Změny neukládejte.</p> <p>Po splnění úkolu minimalizujte okno <i>ModelBuilder</i> a stiskněte F11.</p>
3.	<p>Do diagramu (<i>03_ukol_buffer_parameter</i>) přidejte další prvek: vzdálenost bufferu nastavte jako parametrickou proměnnou.</p> <p>Změny neukládejte.</p> <p>Po splnění úkolu minimalizujte okno <i>ModelBuilder</i> a stiskněte F11.</p>
4.	<p>V toolboxu <i>Diagramy_interaktivita</i> vytvořte nový model (4_prijmeni). Následně sestavte diagram, který provede ořezání bodové vrstvy ORP polygonovou vrstvou ZlinskyKraj.</p> <p>Výstupní vrstvu pojmenujte ORP_prijmeni.</p> <p>Hotový diagram uložte a nechte proběhnout. Po splnění úkolu stiskněte F11.</p>
5.	<p>V toolboxu <i>Diagramy_interaktivita</i> vytvořte nový model (5_prijmeni). Následně sestavte nový diagram pro vytvoření nové souborové geodatabáze (prijmeni.gdb).</p> <p>Umístění nové GDB do adresáře: \diagramy_data</p> <p>Název nové GDB: prijmeni</p> <p>Nakonec nastavte Název GDB jako proměnnou.</p> <p>Hotový diagram uložte a nechte proběhnout. Po splnění úkolu stiskněte F11.</p>

Tab. 2 Zadání úkolů druhého testování interaktivní práce metodou eye-tracking

Úkol	Zadání
1.	<p>Diagram (<i>01_ukol_dissolve</i>) upravte tak, aby mohla proběhnout generalizace (rozpuštění) hranic obcí na hranici kraje. Využijte atribut <i>KRAJ</i>. Změny neukládejte. Po splnění úkolu minimalizujte okno <i>ModelBuilder</i> a stiskněte F11.</p>
2.	<p>Do diagramu (<i>02_ukol_dem</i>) přidejte nástroj, který vytvoří z vrstvy digitálního modelu reliéfu (<i>DMR</i>) vrstvu orientace svahu. Název výstupní vrstvy: <i>orientace</i> Změny neukládejte. Po splnění úkolu minimalizujte okno <i>ModelBuilder</i> a stiskněte F11.</p>
3.	<p>Z daných prvků vytvořte diagram (<i>03_ukol_iterator</i>), který vytvoří kolem vstupní vrstvy železnic obalové zóny (buffery) ve vzdálenosti 1, 2, 3, 4 a 5 kilometrů. Změny neukládejte. Po splnění úkolu minimalizujte okno <i>ModelBuilder</i> a stiskněte F11.</p>
4.	<p>Do diagramu (<i>04_ukol_gdb</i>) přidejte dva další prvky: umístění osobní GDB a jméno osobní GDB jako parametrické proměnné. Umístění nové GDB do adresáře: <i>\diagramy_data</i> Název nové GDB: <i>prijmeni.mdb</i> Změny neukládejte, ale spusťte diagram (Run). Po splnění úkolu minimalizujte okno <i>ModelBuilder</i> a stiskněte F11.</p>
5.	<p>V toolboxu <i>Diagramy_interaktivita2</i> vytvořte nový model (<i>prijmeni</i>). Následně sestavte diagram, který provede vymazání (Erase) polygonové vrstvy Bažiny z polygonové vrstvy Lesy (vrstvy použijte z okna <i>Table Of Contents</i>). Název výstupní vrstvy: <i>Lesy_prijmeni</i> Hotový diagram uložte a nechte proběhnout. Po splnění úkolu stiskněte F11.</p>

Tab. 3 Změna vstupu a výstupů na parametrické v 1. úkolu prvního testování interaktivní práce [s od začátku]

Respondent	Vstup	Výstupy
P02	7	24; 25; 27
P03	8	-
P04	5	7; 9; 12
P05	7	10; 12; 14
P06	5	7; 12; 14
P07	5	8; 10; 12
P08	4	8; 10; 12
P09	7	11; 13; 15
P10	5	11; 13; 15
P11	7	9; 15; 20

Tab. 4 Tlačítko *Auto Layout* v 2. úkolu prvního testování interaktivní práce [s od začátku]

Respondent	Použito	Spatřeno
P02	8	8
P03	-	-
P04	4	2
P05	4	4
P06	-	17
P07	6	1; 3; 5; 14
P08	4	3
P09	6	3; 5
P10	14	14
P11	54	45

Tab. 5 Proměnná v 3. úkolu prvního testování interaktivní práce [s od začátku]

Respondent	Spatřena volba <i>Make Variable</i>	Přidána proměnná pomocí <i>Make Variable</i>	Proměnná změněna na parametrickou
P02	4	10	-
P03	5; 11; 19; 22; 24; 35; 68; 73	38; 81	-
P04	4	8	19
P05	-	-	x
P06	13; 18; 57	-	x
P07	7; 18	16	34
P08	5; 119; 143	148	155
P09	3	10	-
P10	12	19	-
P11	6	17	24

Tab. 6 Přidání prvků ve 4. úkolu prvního testování interaktivní práce [s od začátku]

Respondent	Vstupní vrstva <i>ORP</i>	Ořezová vrstva <i>KRAJ</i>	Nástroj <i>Clip</i>
P02	35	33	74
P03	43	50	109
P04	24	56	33; 56
P05	25	28; 53	37
P06	26	30	72
P07	33	31	43
P08	45	48	70
P09	34	36	75
P10	26	33	54
P11	33	30	60

Tab. 7 Spatření sad nástrojů v okně *ArcToolbox* a přidání nástroje *Create File GDB* v 5. úkolu prvního testování interaktivní práce [s od začátku]

Respondent	Spatřena sada nástrojů <i>File Geodatabase</i>	Spatřena sada nástrojů <i>Workspace</i>	Přidán nástroj
P02	x	x	48
P03	-	-	-
P04	32; 52; 66; 70; 79	31; 35; 75; 102	109
P05	-	-	-
P06	x	x	40
P07	24; 63; 69; 71; 97; 101; 119; 120	107	150
P08	35; 39	40	47
P09	87; 88; 97; 99	93; 116	121
P10	x	x	70
P11	-	25; 27; 29; 49; 51; 55; 56; 63; 65	156

Tab. 8 Nastavení *Dissolve_Field(s)* v 1. úkolu druhého testování interaktivní práce [s od začátku]

Respondent	Spatřen výběr <i>Dissolve_Field(s)</i>	Vybrán atribut <i>KRAJ</i> jako <i>Dissolve_Field(s)</i>
P04	20; 22; 25; 35; 72; 78; 85	-
P05	-	x
P06	-	x
P07(08)	34; 40	36
P09(10)	23; 43; 44; 50	25
P11	23; 50	37
P12	12; 15	-
P13	22; 32; 35	37
P14	15; 25	27
P15	-	x

Tab. 9 Úkol č. 2 druhého testování interaktivní práce [s od začátku]

Respondent	Přidán nástroj <i>Aspect</i>	Spojena vrstva <i>DMR</i> s nástrojem <i>Aspect</i>	Výstup pojmenován <i>orientace</i>
P04	34	43	59
P05	38	44	-
P06	18	24	43
P07(08)	51	57	73
P09(10)	130	143	150
P11	12	22	-
P12	16	22	47
P13	-	x	x
P14	19	30	56
P15	-	x	x

Tab. 10 Úkol č. 3 druhého testování interaktivní práce [s od začátku]

Respondent	Spojen vstup s nástrojem <i>Buffer</i>	Spojena proměnná <i>Distance</i> s nástrojem <i>Buffer</i>	Nastaveny hodnoty v iterátoru	Přidána proměnná <i>%Distance%</i> do názvu výstupu
P04	22	-	-	-
P05	-	-	-	-
P06	12	-	85; 102	148
P07(08)	44	68	20	129
P09(10)	30; 130	77	-	-
P11	13	-	63	-
P12	11; 68	-	38	-
P13	105	-	185	-
P14	13	25	52; 73	100
P15	24	-	75	-

Tab. 11 Proměnná *File GDB Name* ve 4. úkolu druhého testování interaktivní práce [s od začátku]

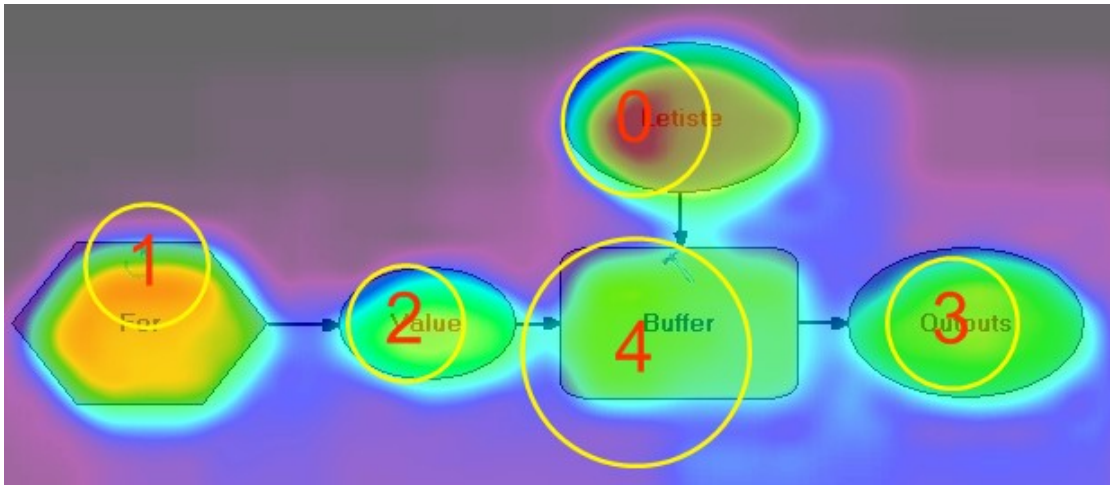
Respondent	Spatřena	Přidána	Změněna na parametrickou
P04	-	-	x
P05	-	-	x
P06	-	-	x
P07(08)	37	38	42
P09(10)	-	-	x
P11	9	16	23
P12	46	62	-
P13	-	-	x
P14	-	-	x
P15	9	-	x

Tab. 12 Přidání prvků v 5. úkolu prvního testování interaktivní práce [s od začátku]

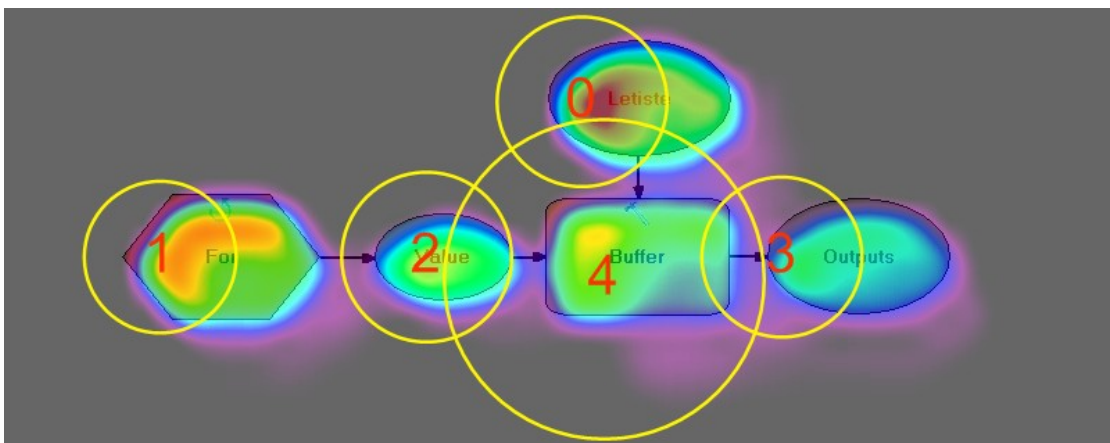
Respondent	Vstupní vrstva <i>Lesy</i>	Překryvná vrstva <i>Bažiny</i>	Nástroj <i>Erase</i>
P04	39	42	52
P05	24	23	36
P06	44	63	35
P07(08)	58	56	42
P09(10)	60	60	25
P11	10	7	54
P12	50	45	36
P13	102; 176	122	76
P14	106	106	31
P15	41; 109	69; 109	51

Příloha 3

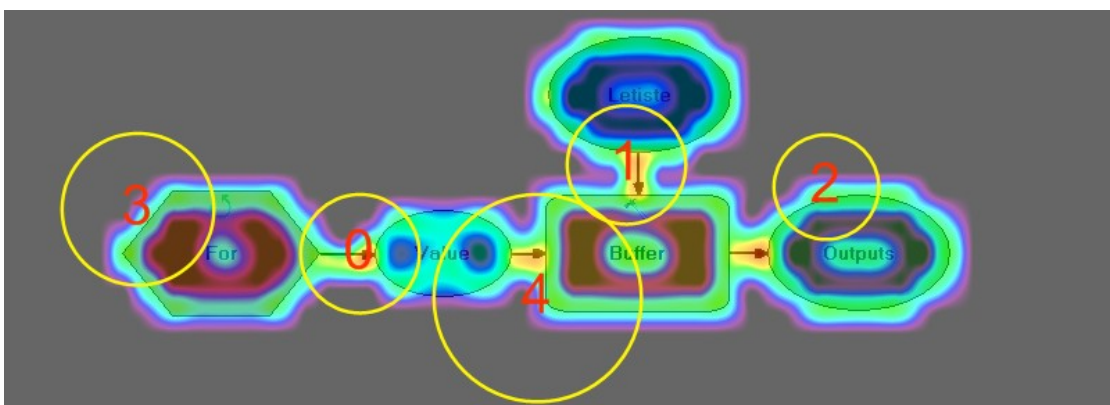
Ukázky saliency map pro předdefinované a vlastní kombinace kanálů



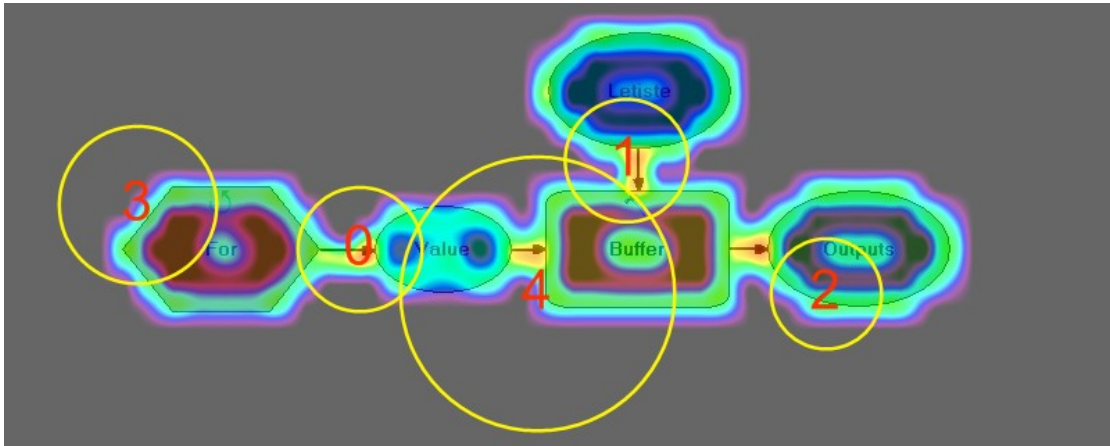
Obr. 13 Saliency mapa s pořadím fixací vypočítána s nastavením předdefinované kombinace kanálů *Std: use all standard channels with unit weights* (ModelBuilder, výpočet OGAMA).



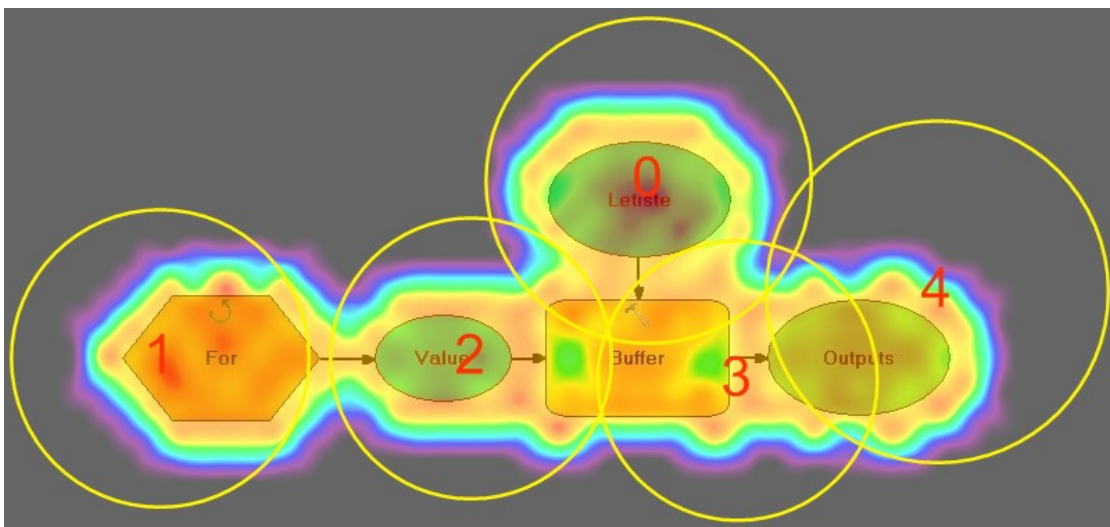
Obr. 14 Saliency mapa s pořadím fixací vypočítána s nastavením předdefinované kombinace kanálů *SurpStd: use all standard Surprise channels* (ModelBuilder, výpočet OGAMA).



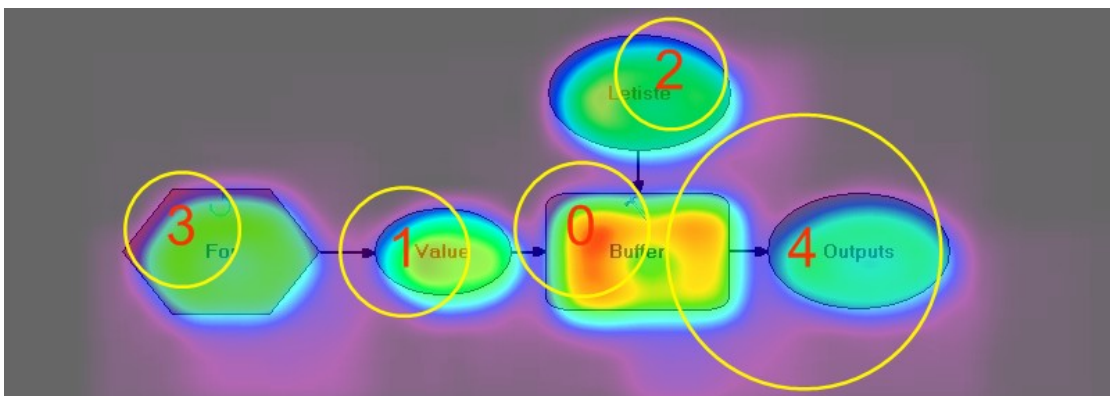
Obr. 15 Saliency mapa s pořadím fixací vypočítána s nastavením předdefinované kombinace kanálů *PN03contrast: Parkhurst & Niebur'03 contrast model* (ModelBuilder, výpočet OGAMA).



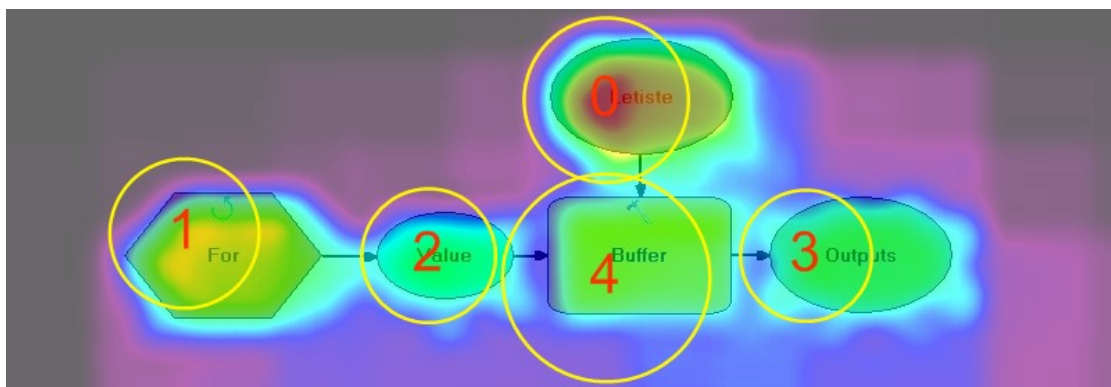
Obr. 16 Saliency mapa s pořadím fixací vypočítána s nastavením předdefinované kombinace kanálů *Variance: local variance in 16x16 image patches* (ModelBuilder, výpočet OGAMA).



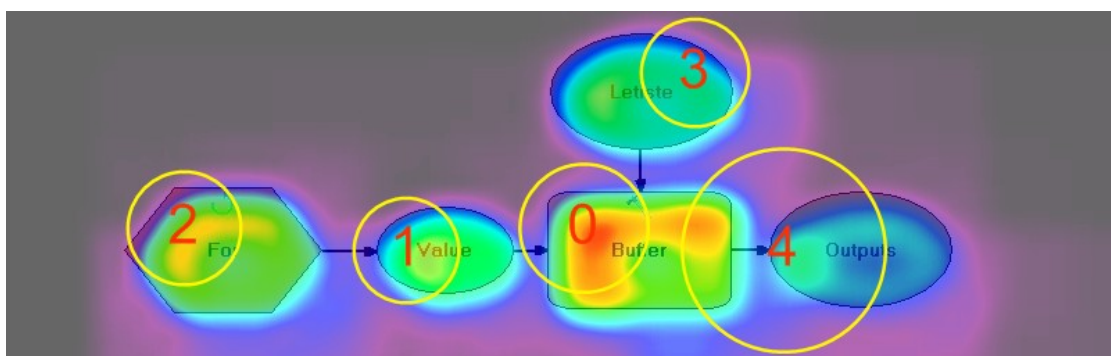
Obr. 17 Saliency mapa s pořadím fixací vypočítána s nastavením předdefinované kombinace kanálů *Scorr: spatial correlation between image patches in a frame* (ModelBuilder, výpočet OGAMA).



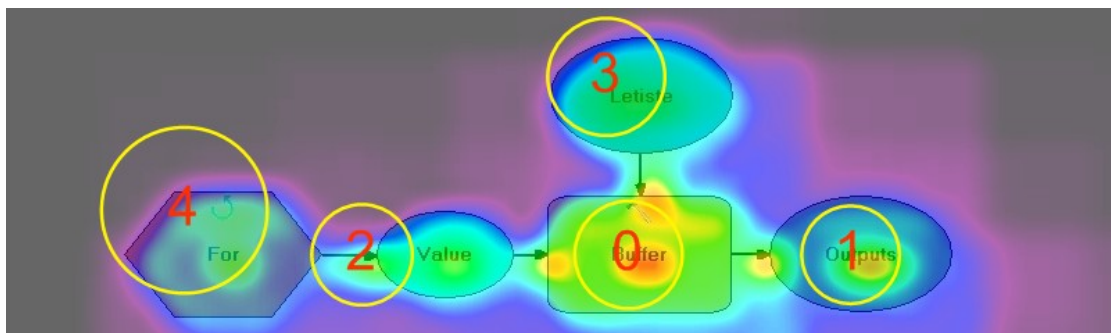
Obr. 18 Saliency mapa s pořadím fixací vypočítána s nastavením kanálu *Color (S)* a váhou 1 (ModelBuilder, výpočet OGAMA).



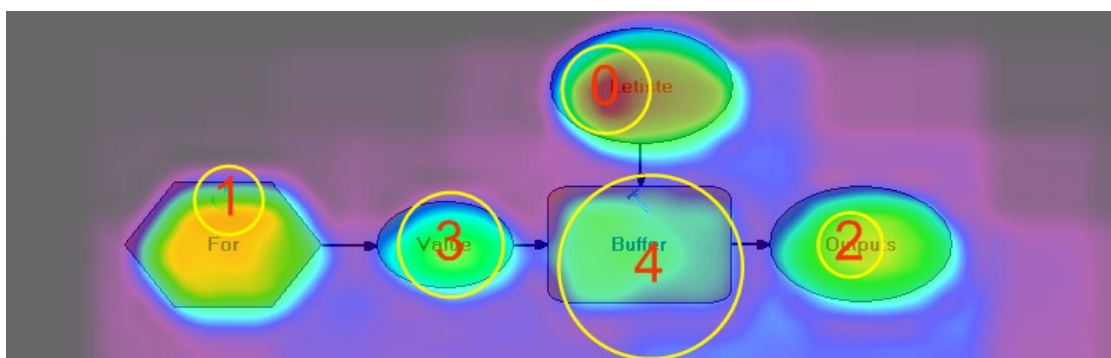
Obr. 19 Saliency mapa s pořadím fixací vypočítána s nastavením kanálu Color (G) a váhou 1 (ModelBuilder, výpočet OGAMA).



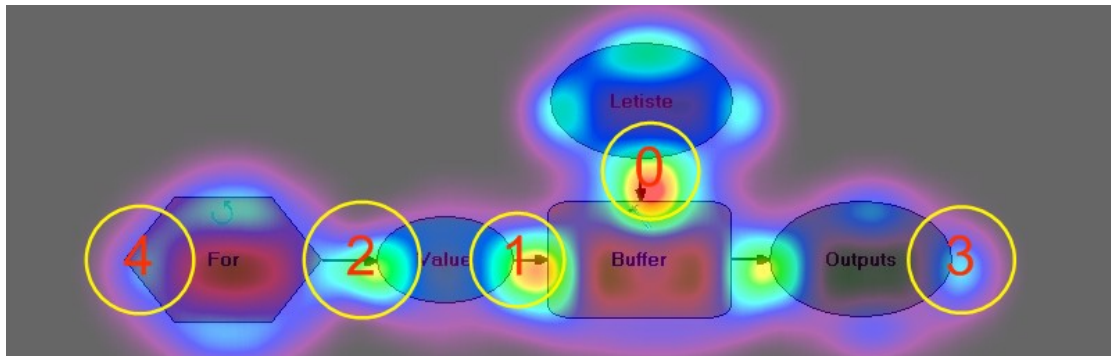
Obr. 20 Saliency mapa s pořadím fixací vypočítána s nastavením kanálu Color (C) a váhou 1 (ModelBuilder, výpočet OGAMA).



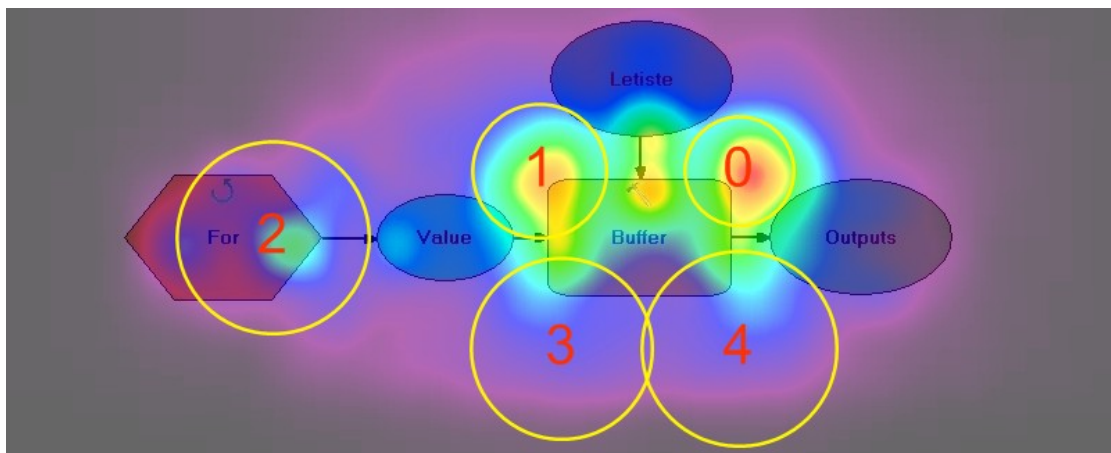
Obr. 21 Saliency mapa s pořadím fixací vypočítána s nastavením kanálu Intensity (N) a váhou 1 (ModelBuilder, výpočet OGAMA).



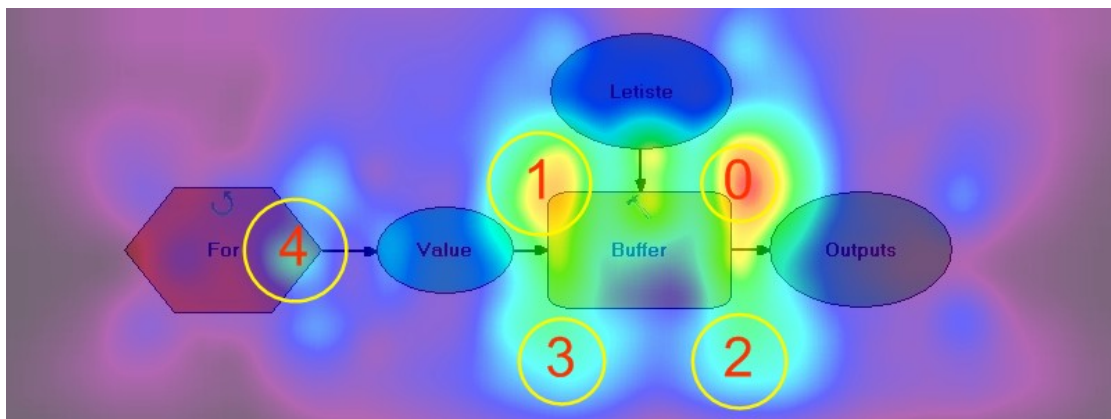
Obr. 22 Saliency mapa s pořadím fixací vypočítána s nastavením kanálu Intensity (I) a váhou 1 (ModelBuilder, výpočet OGAMA).



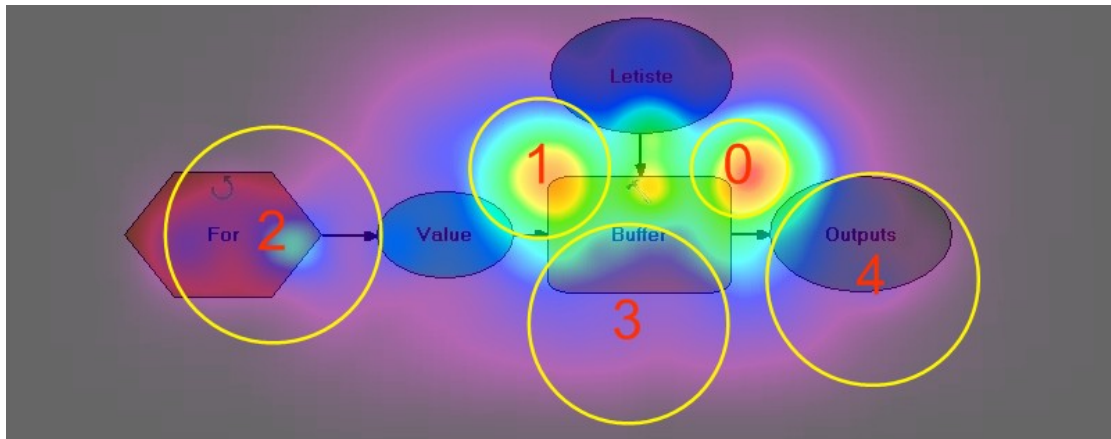
Obr. 23 Saliency mapa s pořadím fixací vypočítána s nastavením kanálu Orientation a váhou 1 (ModelBuilder, výpočet OGAMA).



Obr. 24 Saliency mapa s pořadím fixací vypočítána s nastavením kanálu T-junction detector a váhou 1 (ModelBuilder, výpočet OGAMA).



Obr. 25 Saliency mapa s pořadím fixací vypočítána s nastavením kanálu L-junction detector a váhou 1 (ModelBuilder, výpočet OGAMA).



Obr. 26 Saliency mapa s pořadím fixací vypočítána s nastavením kanálu X-junction detector a váhou 1 (ModelBuilder, výpočet OGAMA).