Research Articles in Simplified HTML: a Web-first format for HTML-based scholarly articles

Silvio Peroni, Francesco Osborne, Angelo Di Iorio, Andrea Giovanni Nuzzolese, Francesco Poggi, Fabio Vitali and Enrico Motta

ABSTRACT

- 目的.介绍了RASH，以及与之伴随的RASH框架
- 设计.RASH设计的目标是：易于学习和使用；可通过网络分享学术文档；可在现有的出版流程中被采用
- 发现.RASH适合被研讨会、会议以及期刊等采用，而且熟悉HTML的研究者也能轻松学习它
- 研究局限.对于不那么懂技术的研究者可能采用可能有一定难度；而且还需要更多工具的开发，比如用来同openXML格式的转换
- 现实意义.RASH及其框架向实现论文内容意义的形式化表示又迈出了一步，促进了论文的自动化发现，使之链接到语义相关的文章，以可操作的形式提供了对文章中数据的访问，并允许集成论文之间的数据

ABSTRACT

- 社会意义.RASH解决了学术论文的不同主体间的内在需求：研究者(关注其内容)，读者(体验浏览它的新方式)，公民科学家(通过语义标注重用其可用数据)，出版商(使用新技术的优势)

- 价值.帮助作者专注于文本的组织、以语义化的方式丰富文章内容以及将验证、可视化、转换等问题都交由RASH框架来处理

Introduction

- 2014年底，web技术邮件列表的一些帖子以及语义网社区讨论了一些学术交流中的“长青”话题，比如作者怎么才能以HTML格式提交他们的论文而非PDF、MS Word或者LaTeX呢？
- Web-first格式研究论文指的是以HTML来设计、存储和传输，学术界对于它的兴趣一直在上升。举例说两个研究成果：Scholarly HTML和Dokiely。
- 采用HTML不仅可以简化和统一起草、提交、和出版时的数据格式，还有一个潜在的原因是HTML的采用可以方便语义标注的嵌入，从而促进研究交流。

What is RASH?

- 概述：RASH是提供给web-first论文的作者一组HTML的子集，这种格式仅仅包含32个HTML元素。RASH还伴随着RASH Framework，框架提供了一些规范和工具。
- 特殊性：相比于其他类似格式，RASH的特殊性主要表现在两点：一是RASH采用了简化的基于模式的数据模型（simplified pattern-based data model）。标注元素被减少到最低。二是RASH并没有给出一个新的创作环境，而是以word 、ODT和LaTeX等编辑软件为基础。
- 目的：初衷是为了帮助作者专注于文本组织并支持他们对论文内容进行语义增强，花更少的功夫在思考论文的视觉呈现上。这印证了语义出版中的一个原则：关注点分离（clear separation of concerns）。

ARTICLE STRUCTURE

3

Which ``Web-first'' Format for Research Articles?

介绍了创造一种新的学术出版的网络优先格式的原理，并讨论了最小化的重要性

5

RASH Framework

介绍了RASH的理论背景，并且简单说明了RASH的语言和工具使用

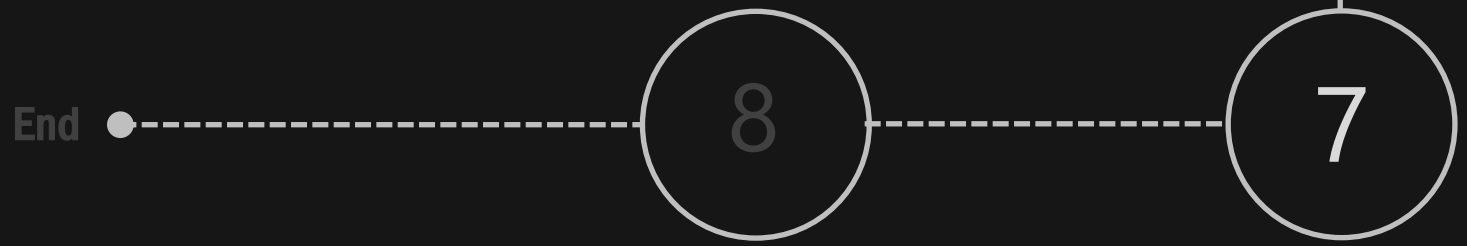
4

Writing Scholarly Articles in HTML with RASH

6

RASH and SAVE-SD: an Evaluation

对RASH在SAVE-SD 2015和2016两年的应用情况的总结



Acknowledgements

Conclusion

对论文的总结和对未来发展的概述

学术界对于web-first格式的研究论文一直很有兴趣，因而近年有一些同RASH相关的研究成果涌现。文章主要把它们分成两类，一类是基于HTML的格式，第二类是HTML文档的WYSIWYG编辑器。

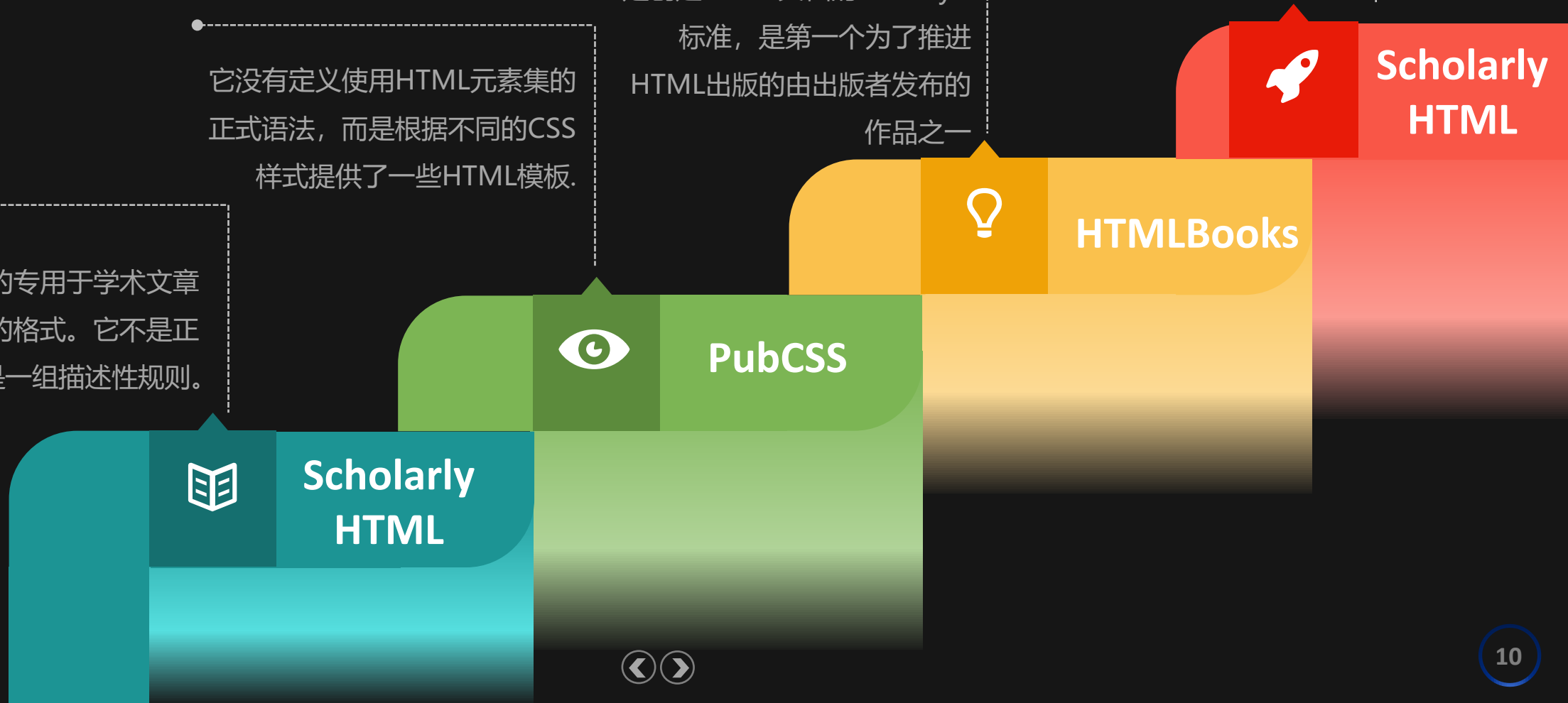
（1）HTML-based formats

是第一次提出的专用于学术文章的基于HTML的格式。它不是正式的语法，而是一组描述性规则。

它没有定义使用HTML元素集的正式语法，而是根据不同的CSS样式提供了一些HTML模板.

是创建HTML文档的O'Reilly's标准，是第一个为了推进HTML出版的由出版者发布的作品之一

提供一种基于开放标准的特定领域的数据格式



(2) HTML-oriented WYSIWYG editors

Dokiely

是允许在浏览器上直接创作学术文章的应用，并实现了以RDF标注和通知的一些特性

Fidus Writer

是一个通过类似文字编辑器界面的用于创作HTML学术文档的基于网络的应用

Authorea

是一个允许用户通过清楚高效界面写论文的网络服务

dokieli

Fiduswriter

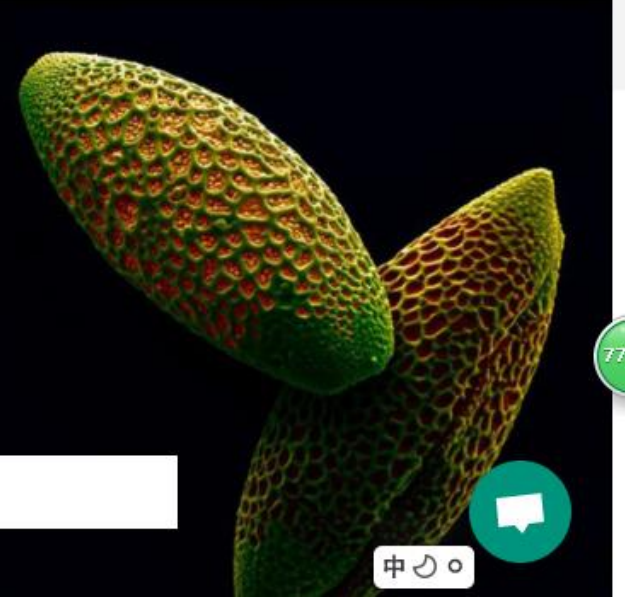
CONTRIBUTE WHAT IS IT? FAQ FORUM DOWNLOAD

AUTHOREA

PRODUCT EXPLORE ABOUT PRICING LOG IN SIGN UP

Writing documents for the 21st century

Write, manage, and publish your documents alongside 100,000+ writers and researchers.



NAME E-MAIL ADDRESS PASSWORD

Yes! Send me a monthly newsletter.

正在连接...

WHICH “WEB-FIRST” FORMAT FOR RESEARCH ARTICLES?

“Web优先”格式表明了使用HTML作为主要格式来编写、存储和传输论文的可能性，而不仅仅是在Web上提供这些文章。在这种情况下自然会出现一些问题:我们应该使用完整的HTML语言吗? 如果只使用一个有限的子集，我们应该考虑哪些元素? 我们是否需要对语言的使用有明确的规定?

认为应使用完整HTML语言

- 有些论文认为应允许作者使用任何他们想要的HTML结构来撰写论文，这将减少甚至消除对*模板瓶颈（template bottleneck）*的恐惧。

认为应对HTML加以限制

- 让作者自由使用整个HTML规范可能会在某种程度上影响文章的整个写作和发布过程。这种自由能导致两类主要问题，一是可视化瓶颈，二是作者会较少关注研究内容。
- 可以使合作论文的写作变得更加方便，避免不同作者间互不理解的情况。
- 有利于和其他复杂格式间的相互转换。在完整的HTML中，转换要更复杂、容易出错并且不那么精确。
- 对出版者而言，允许作者随意以自己的方式使用HTML可能会降低其工作效率。
- 更适合语义出版的应用。如果作者以不同的方式使用HTML，那么从中提取文本的标记结构则非常困难。

WRITING SCHOLARLY ARTICLES IN HTML WITH RASH

RASH中提出的HTML子集严格遵守了*模式理论(patterns theory)*。模式是针对反复出现的问题的一个被广泛接受的解决之道。因此接下来主要介绍了文档设计的模式，然后详细说明了RASH的各个细节。

Theoretical foundations: structural patterns

1. inline [+t+s+T], e.g., the element em;
2. block [+t+s−T], e.g., the element p;
3. popup [−t+s+T], e.g., the element aside;
4. container [−t+s−T], e.g., the element section;
5. atom [+t−s+T], e.g., the element abbr;
6. field [+t−s−T], e.g., the element title;
7. milestone [−t−s+T], e.g., the element img;
8. meta [−t−s−T], e.g., the element link.

模式理论背后的基本观点是标记语言的每个元素应该只遵循唯一一种结构化的模式

取决于元素：能否包含文本（+t、-t）;能否包含其他元素（+s、-s）;所包含它的元素能否包含文本（+T、-T）

结合这些所有可能的值，我们基本上可以得到8种核心的结构模式

PATTERNS THEORY

这个理论的观点是，只需很少的的结构模式就足以表达大多数用户定义文档组织的需求

正交性

每个模式都有一个特定的目标，并且适合特定的上下文

可组装性

每个模式只能在其他模式的某些上下文中使用

RASH: Research Article in Simplified HTML

RASH将HTML元素的使用限制到了仅仅32个，而且允许作者嵌入RDF标注。除此之外，RASH还严格遵守了数字出版的 WAI-ARIA Module 1.0来表示结构化语义。

Development and patterns

RASH的发展始于整个HTML5语法，然后通过消除和限制特定的HTML元素的使用,让他们能够表达代表学术论文的结构并让语言完全符合XML文档的结构模式理论。

Table 2 **The use of structural patterns in RASH.**

Pattern	RASH element
inline	a, code, em, math, q, span, strong, sub, sup, svg
block	figcaption, h1, p, pre, th
popup	*none*
container	blockquote, body, figure, head, html, li, ol, section, table, td, tr, ul
atom	*none*
field	script, title
milestone	img
meta	link, meta

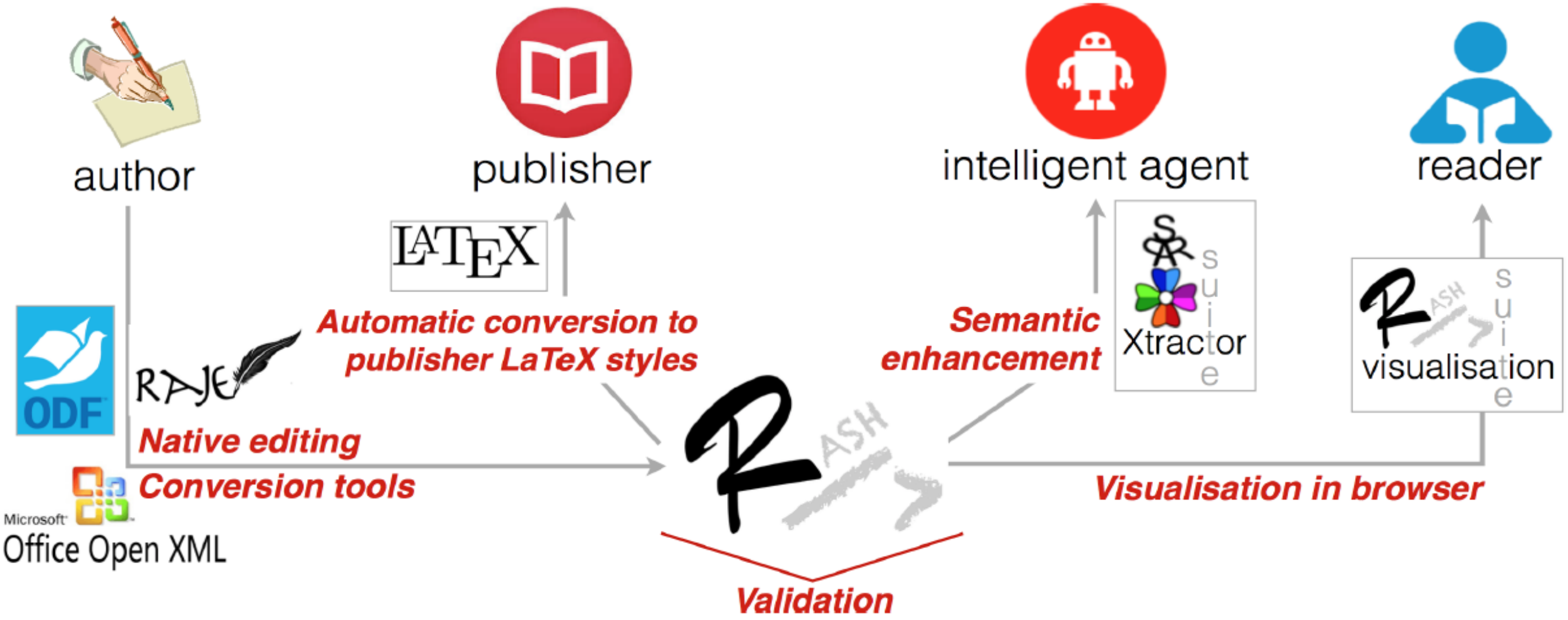
Development and patterns

- 要注意的是，RASH没有用到atom和popup这两种模式。
- 符合atom模式的元素被包含在离散的模块中（e. g. paragraphs），并且只包含文本而且没有任何额外的元素。这在学术写作中是不常见的，因为元素往往能包含其他元素（e.g.an emphasis can contain a link）。
- Popup本来是用来代表插入但并不打断正文行文的复杂结构的，比如说脚注。符合popup模式的元素，可以在具有混合上下文的元素[+t+s]中被发现。

THE RASH FRAMEWORK

当提出了一种新的标记语言，则需要提供以RASH写文章的工具。不可否认的是，不是所有的作者 都可以或者说愿意以HTML写文章和手动地添加语义标注。因此便提出了RASH框架，它是为论文写作提供的一组规范以及写作、转换、提取的工具。

THE RASH FRAMEWORK



THE RASH FRAMEWORK

验证

为了检查一个文档是否符合RASH规范，框架中开发了一个脚本，使用户能够检查文档，使之同时符合RASH的RelaxNG语法和HTML规范。

Dirt Kiyut
Marketing 1

可视化

文档的可视化由合适的CSS3样式表和JavaScript通过浏览器呈现。

Rafael De Kuib
Marketing 2

转换

开发了一个将RASH文档转换成不同的LaTeX样式的XSLT文档。还开发了XSLT 2.0文档，用于由ODT和DOCX文档制作RASH文档。

Bann Dempo
Salesman 1

ROCS

创造了一个名为ROCS（RASH Online Conversion Service）的网上转换工具，用于支持作者RASH文档的写作和提交，以便能够方便地被期刊、研讨会和会议加工。ROCS结合了前面提到的几种工具。

Dul Gembuk
Salesman 2

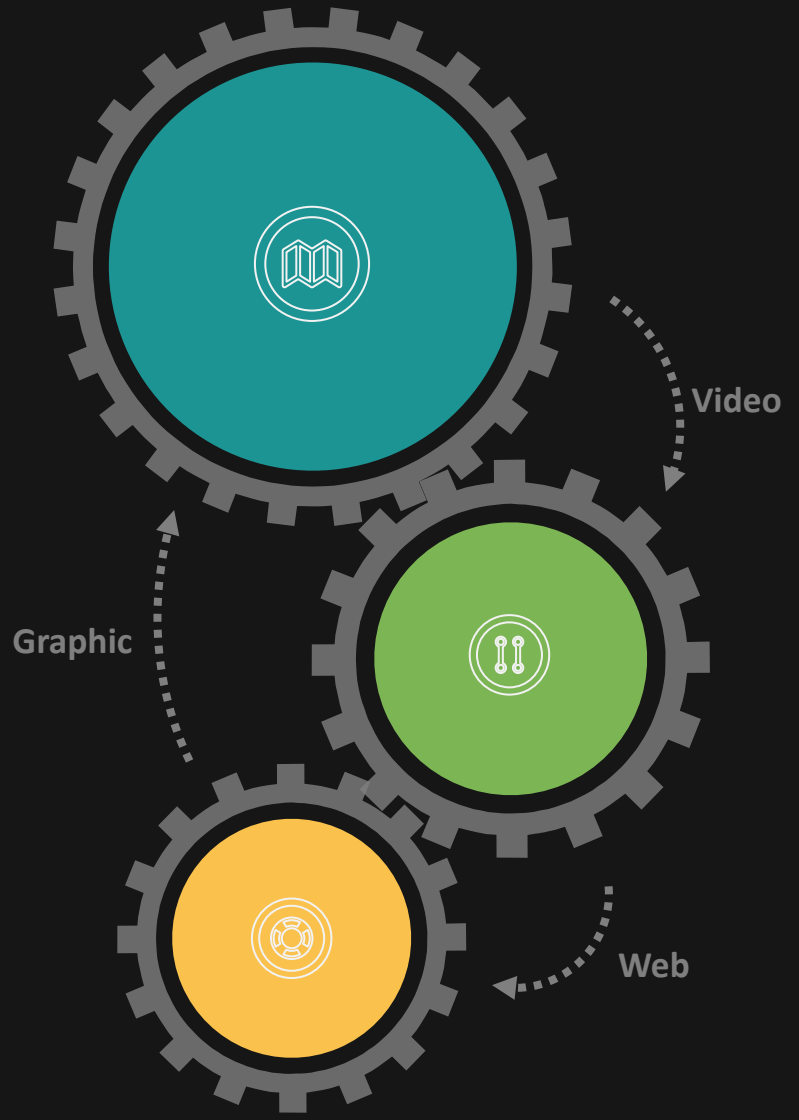
THE RASH FRAMEWORK

以结构化语义丰富RASH文档

框架的另一个开发涉及到使用RDFa标注自动丰富RASH文档，根据FaBIO和DoCO定义了这些文档的实际结构，开发了一个名为SPAR Xtractor suite的Java应用程序。它被设计为一键式工具，可以为RASH文档添加自动的结构化语义.

用本地编辑器写RASH文档

RASH近期的一个开发是RAJE（RASH Javascript Editor），它是一个WYSIWYG的平台。RAJE向它的用户保证文字处理器与基于html的格式相结合的优点，即交互性、可及性、易于机器处理。



RASH AND SAVE-SD: AN EVALUATION

对RASH的真正验证在于其被作者 、 研讨会应用以及其在发布过程中的效果。 因此在评估这一节主要展示了基于来自SAVE-SD 2015和2016两年的作者和评估者完成的使用RASH后的体验结果的分析，以及对RDF标注在相关论文中的研究。

USER BACKGROUND

2015年，作者主要来自语义网社区，因此对例如RDFa和Turtle等技术都很熟悉。他们中的大多数都知道怎么正确标注HTML文档和理解在论文中包含语义关系的好处。他们也更常用LaTex而不是word或openoffice来写论文。2016年，情况有所改变。只有57%的使用者对语义技术很熟悉。说明RASH开始引起具有不同研究背景的、不怎么懂技术的研究者的兴趣。

Table 3 User background for SAVE-SD 2015, SAVE-SD 2016.

Year	MS Word	OO Writer	LaTeX	HTML	XML	RelaxNG	SW	RDFa	Turtle	JSON-LD
2015	33%	33%	83%	83%	100%	67%	83%	100%	100%	50%
2016	57%	0%	71%	71%	71%	29%	57%	57%	57%	43%
AVR	40%	13%	67%	67%	73%	40%	60%	67%	77%	40%

RASH usability

- SUS（系统可用性测量）对RASH的可用性进行了定量分析。得出的分数尽管不是特别高，但也是能接受的。
- RASH的平均分是62.7±11.9，略低于SUS的平均分。然而根据个人背景的不同，SUS的得分有很大的差异。
- 在LaTeX和SWT方面有较强专业经验的用户比其他作者获得了显著更好的SUS分数，而具有HTML专业经验的作者表现出的优势则并不是很大

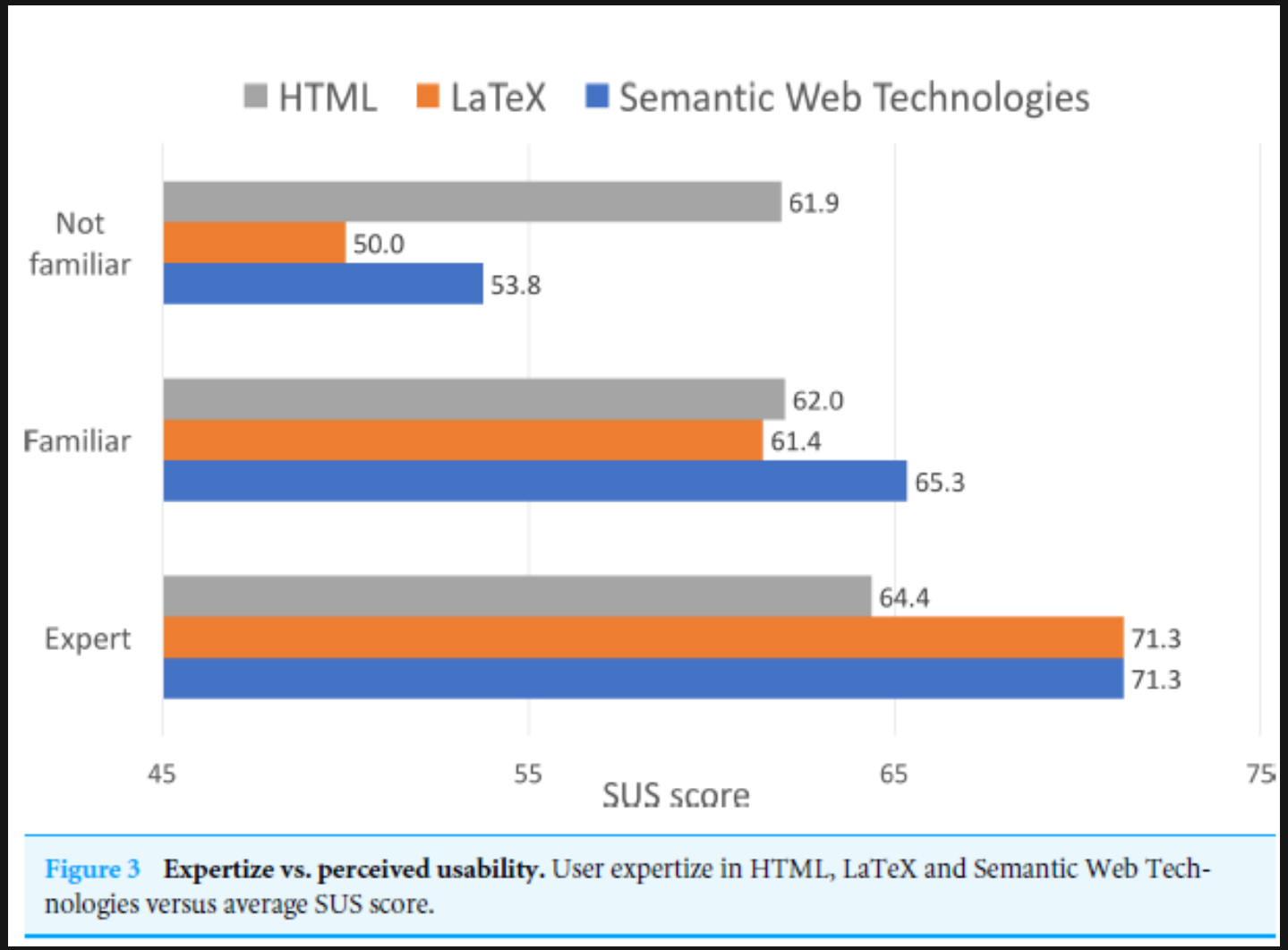


Figure 3 **Expertize vs. perceived usability.** User expertize in HTML, LaTeX and Semantic Web Technologies versus average SUS score.

CONCLUSIONS

- RASH是作为HTML子集的一种标记语言，用于撰写科学论文。
- RASH Framework是用于以RASH语言撰写论文的一套规范和工具。
- 文章还讨论了RASH开发背后的基本原理，并且展示了目前这种语言的验证、可视化、转化、提取、编辑等各种工具。
- 文章还通过两届SAVE-SD评估了RASH的适用性和潜力。据我们所知，这是第一次对采用基于html的语言编写科学论文进行实证评估。评估显示了RASH可以成功地应用于研讨会和会议，表现出了作者良好的接受性和同出版过程顺利的结合。
- 随着不久的将来的发展，我们计划开发工具来使RASH文档的语义丰富过程自动化。例如，我们目前正在研究章节修辞学和引文功能的自动识别，以便根据两个SPAR本体，即DoCO和来描述它们。
- 我们打算进一步发展RASH Framework。首先，我们正在开发更精细的创作工具和转换器。

dokieli: decentralised authoring, annotations and social notifications

http://csarven.ca/dokieli

When I saw the computer, I said, ‘at last we can escape from the prison of paper’, and that was what my whole hypertext idea was about in 1960 and since. Contrarily, what did the other people do, they imitated paper, which to me seems totally insane - Ted Nelson.

dokieli是一个通用的客户端应用程序，用于文档创作、发布和交互。该工具的功能是根据用户的需求和技术储备来设计的。该编辑器建立在open Web standards之上，而且文档符合linked data最佳实践。这篇论文就是一个dokieli实例。

一个持久的系统必须是可进化的。我们很难预测哪种技术会随着时间的推移而持续下去，但是我们从Evolution of the Web中获得灵感，追求简单性、灵活性、分散性、互操作性和容忍度等特性。

ARCHITECTURE

01 Data format

02 Identifiers

03 Vocabularies

04 Distribution

05 Data storage

06 Modes of operation

DATA FORMAT

- dokieli文档的本机序列化格式是是HTML+RDFa。HTML支持人类可读文章，RDFa则能很好地适用于dokieli。RDFa允许作者为其文章的内联想法增加语义结构。

- Turtle, JSON-LD, and TriG之类的替代语法可能作为原始数据嵌入到HTML中，这会导致不必要的分离，并可能导致数据的重复和不同步。

IDENTIFIERS

当dokieli被发布在网上，它们就会有自己的URI。这可以用于明确地引用文档。此外，文档的任何单个单词、短语、段落或其他小节都可以使用片段式的URI，拥有自己的唯一标识符。

VOCABULARY

dokieli不强制使用任何特定的RDF词汇表，因为文章的内容决定了如何最好地描述它。默认情况下，dokieli文档使用以下词汇表，而它们能由作者自行决定是否被替换或添加。

- General-purpose: schema.org
- Publishing and referencing: SPAR Ontologies
- Annotations: Web Annotations
- Social notifications: ActivityStreams and Pingback
- Links to personal storage and user preferences: LDP and Solid
- Access control: WebAccessControl/ACL

Distribution

用JavaScript编写的dokieli应用程序逻辑分配到每个文档之中，以确保所有文档都可以编辑和交互。

Dokieli进步的一个本质特征是其容错性。如果应用程序脚本或样式表不可用，内容仍然是可访问的，而不是完全的崩坏。

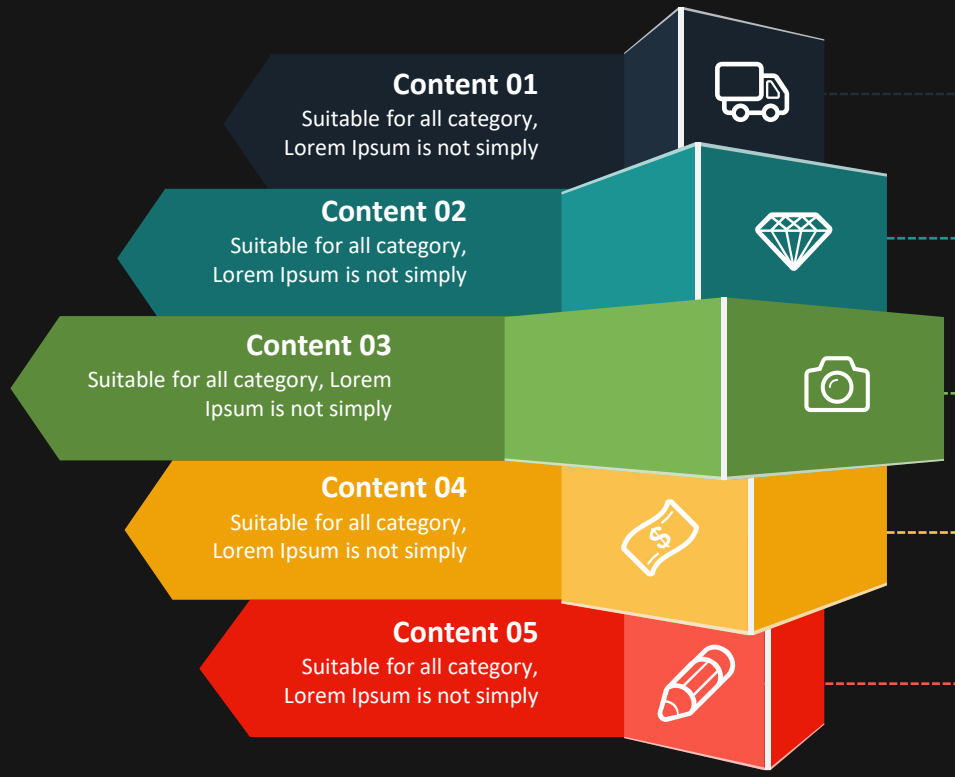
dokieli是自我复制的，因为dokieli文档的阅读器可以在单击按钮时将实例(副本或全新的空文档)衍生到自己的存储空间中。

DATA STORAGE

dokieli文档的所有者拥有完全的控制权，并且在何处存储文章上有很大的灵活性。

MODES OF OPERATION: PUBLISHING AND INTERACTING

Publishing



- Read only
- Temporary edit
- Local persistent edit
- Persistent edit
- Data only

IMPLEMENTATION

IMPLEMENTATION

Semantics and linking

Rich editing

Authentic-ation and access control

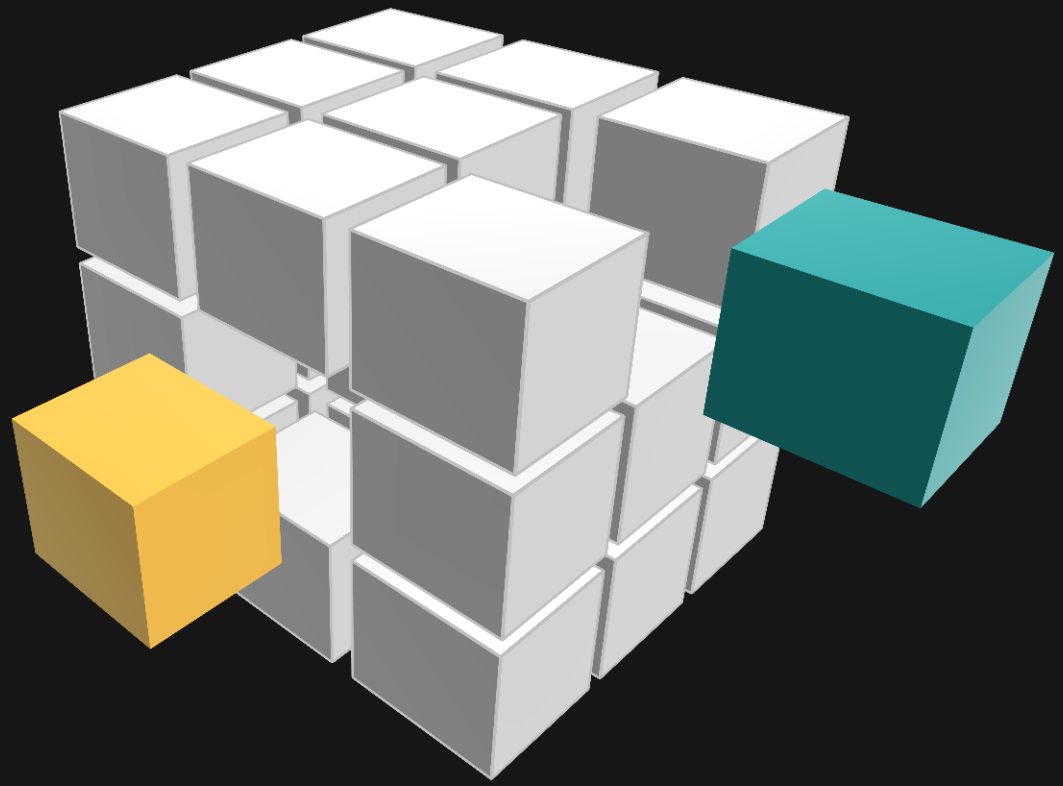
Storage

Interactions

Social and Technical Impact

Data generation

希望dokieli的使用能将以前不常见但非常有价值的链接开放数据汇集到Web中.



Data ownership and reuse

关联数据方法的一个特别优点是，用dokieli生成的数据可以被其他应用程序重用，并且很容易与来自其他数据源的数据集成

THANK YOU FOR WATCHING