TAR

The IPSI BgD Transactions on Advanced Research

Multi-, Inter-, and Trans-disciplinary Issues in Computer Science and Engineering

A publication of IPSI Bgd Internet Research Society New York, Frankfurt, Tokyo, Belgrade
January 2016 Volume 12 Number 1 (ISSN 1820-4511)

Special issue: „Using the Web: New Alternatives“

Guest Editor: Hermann Maurer

Table of Contents:

The IPSI BgD Internet Research Society

The Internet Research Society is an association of people with professional interest in the field of the Internet. All members will receive these TRANSACTIONS upon payment of the annual Society membership fee of €500 (air mail printed matters delivery).

Member copies of Transactions are for personal use only

IPSI BGD TRANSACTIONS ON ADVANCED RESEARCH

www.internetjournals.net

Editorial

Veljko Milutinovic and Jakob Salom, Co-Editors-in-Chief

This special issue treats digital preservation of global heritage as an infrastructure for data mining of hidden knowledge. All the selected papers do offer features that could be used to make the dataming process more effective. This may not be visible at the first glance, but each selected paper does offer mechanisms that could be utilized for various aspects of the general problem.

Also, this special issue also gives a special view of the past activities of Professor Hermann Maurer, a Professor Emeritus of the University of Graz in Austria. His pioneering research in digital heritage left a deep impression on the researchers worldwide and served as the inspiration for many, including the authors of the papers selected for this special issue.

Finally, this special issue could serve as the guideline for those who like both to learn about the past research efforts in this important and continuously emerging field and to obtain the knowledge of the future research and development trends. Prediction of the future could be generated by extrapolating from the past, but also by interpolating in the domains of contributions of those who lead the vision of the field.

The first paper, ***The Next Quantum Jump of the Internet*** (by Vladisav Jelisavcic and Herman Maurer, from Serbia and Austria respectively) describes the concept of online content curation that has emerged with the ever-increasing need for finding and structuring the vast amount of information available on the Web. The authors attempt to define different roles and scenarios of content curation, explain why there is so big a number of curation tools on the Web, and present capabilities different major curation tools have.

In the second article, ***Optical Character Recognition of Old Fonts – A Case Study*** (by Johanna Pirker and Gerhard Wurzinger, both from Austria) the authors explain how challenging it is to extract texts from images of texts especially in historic books. Due to low-quality images, rare fonts, and unknown dictionary standard OCR, software often fails in recognizing these texts. The authors discuss existing OCR systems with a focus on learning strategies, and present an OCR model which is optimized to recognize old books. Additionally, they describe the process to measure the quality of the outcome.

The past few years have been marked by an increased use of sensor technologies, abundant availability of mobile devices, and growing popularity of wearables, which enable a direct integration of their data as part of rich client applications. Despite the potential and added value that such aggregate applications bring, the implementations are usually custom solutions for particular use cases and do not support easy integration of further devices. In the third article, ***Smart Web Services (SmartWS) – The Future of Services on the Web*** (by Maria Maleshkova; Patrick Philipp; York Sure-Vetter; and Rudi Studer, all from Germany) the authors introduce Smart Web Services (SmartWS) that not only provide remote access to resources and functionalities, by relying on standard communication protocols, but also encapsulate ‘intelligence’. Smartness features include, among other, context-based adaptation, cognition, inference and rules that implement autonomous decision logic in order to realize services that automatically perform tasks on behalf of the users, without requiring their explicit involvement. They present key characteristics of SmartWS, introduce a reference implementation framework, describe a specific use case for implementing SmartWS in the medical domain, and specify a maturity model for determining the quality and usability of SmartWS.

The topic of this special issue is digitalization. The fourth and the last paper on this topic ***Digitalization of Scientific and Cultural Heritage in Dissemination of Innovative Teaching Models*** (by Milica Lajbenšperger, Marija Šegan, Sandra Vujošević, and Zoran Ognjanović) analyses digitalization from a totally different view - how the digitization of cultural and scientific heritage can help improve high school and university teaching of mathematics, computer science, astronomy, and history and Serbian language. The paper is a part of a program started by The Mathematical Institute of the Serbian Academy of Sciences and Arts, which goal is digital literacy of teachers and students through the implementation of principles and digitization tools in educational work of high schools. Unlike the past work with big student groups, the authors are now testing the hypothesis of small homogeneous group efficacy. The first results show that the work in small homogeneous groups has motivated the teachers involved in the program to implement more of the innovative teaching models, which in return resulted in a better students' success.

A very interesting fifth article of this issue ***A Dataflow Machine Architecture for Static Dataflow Program Graphs*** (by Lorenzo Verdoscia, from Italy) goes out of the scope of the special issue topic and presents an exciting dataflow machine prototype. Reconfigurability and huge density of today's devices constitute an ideal tool to experiment and implement new forms of computation. The number of cores integrated onto a single die is expected to climb steadily in the foreseeable future. This transition to many-core chips (thousand of cores per processor) is driven by a need to optimize performance per watt. Thus, several computer research groups are trying to understand how to best connect these cores, to design parallel programs, and how many core architectures and their software can scale to the thousands of cores that hardware will be able to support in a decade. In this scenario, the question is whether it is time to revaluate dataflow and functional paradigms. The author suggests a dataflow machine prototype that includes identical processors which constitute the reconfigurable environment devoted to accelerate the execution of chunks of dataflow program graphs directly into hardware. As a result of the co-design approach, the one-to-one mapping between actors of the model and computing units of the processor happens in a straightforward manner. In this way, it is possible the execution of static dataflow program graphs without using memory to store partial results when data tokens flow from a computing unit to another; and without generating control tokens during the computation so that graph executions occurs in a completely asynchronous manner. Finally, by means of an FPGA-based demonstrator, specifically realized to validate the basic design choices of the prototype, some experimental results in solving a linear equation system with the Jacobi and Gauss-Seidel iterative algorithms are presented.

The Next Quantum Jump of the Internet

Jelisavcic, Vladisav and Maurer, Herman

Abstract: The vast amount of information available on the Web is increasing every day. With the ever-increasing need for finding and structuring the information, a concept of online content curation has emerged. In this study, we attempt to define different roles and scenarios of content curation. Recently, there has been a surge of different curation tools available on the Web. The main goal of this paper is thus to define differences in capabilities curation tools can have that made possible the existence of so many diverse tools. We, also, present a tabular comparison of major curation tools characteristics.

Index Terms: ***Curation tools, knowledge portals***

1. Introduction

It is increasingly clear that just about any valuable information is on the Web, yet impossible to find, since the amount of information is getting too large, too redundant and is often of doubtful quality. Because of this, recently the notion of curating information started to emerge, i.e. of aggregating relevant information for specific topics in one way or another. Various approaches have been tried.

(i) Collecting information on special topics on dedicated servers or portals, assuring their quality by verification through experts, as is the case with some medical servers (information is for pay), the portal Europeana [28] trying to tie together the precious holdings of art in European archives and museums, or attempts like the Austria-Forum [21] or Serbia-Forum trying to provide information of interest for certain regions.
(ii) There have been attempts to make a huge body of books available assuming that this guarantees a certain quality or at least the knowledge of the source, like in the open library or the Google books project.
(iii) There has been the attempt of Wikipedia trusting that quality of information can be assured through the “wisdom of crowds”.
(iv) Some “curating services” have tried to collect information of interest with sufficient quality by allowing interested groups of (hopefully) specialists to assemble pointers to contributions of interest. Each approach has its merits, yet none solves the real problem of being able to easily find reliable and deep information on any subject of interest.

V. Jelisavcic, Mathematical Institute of the Serbian Academy of Sciences and Arts, vladisav@mi.sanu.ac.rs
H. Maurer, Graz University of Technology and Academia Europaea, hmaurer@iicm.edu

2. Classification of Curation Tools

2.1 Reasons for Curation

There are many reasons that make content curation develop so fast these days; here we will focus on the following three:

1. Information synchronization
2. Marketing (eCommerce)
3. Knowledge dissemination (eEducation)

There are several settings for content curation, each one corresponding to a different goal to be accomplished. Each type of curation has different priorities and different types of users; however, the underlying cause is mainly the same - collection and transfer of the information and/or knowledge. We will try to define three main curation settings corresponding to the reasons for curation that we explained earlier:

1. Content is curated for someone to stay in touch with new information;
2. Content is curated to promote a product or an idea;
3. Content is curated for someone to learn something;

In the first setting, the author of curation is typically an expert on some topic, or a journalist (or a blogger), while curation users are typically persons who are at least broadly interested or engaged in the content topic (personally or professionally), seeking the newest and the most recent information. Here, the curation author acts as a journalist, a curation user is a reader, and the goal of curation is information sharing.

In the second setting, the author of curation is typically a marketing expert targeting the audience that might find value in the content topic. Curation users can be the same as in the first setting, but also include users who are simply looking for fun. Here, the curation author acts as an entertainer, curation user is an audience, and the goal of curation is story telling (or marketing).

In the third setting, the author of curation is typically an expert on a topic, but the curation users are typically users hoping to gain knowledge. Content curated in this setting allows faster learning curves, and even access to otherwise inaccessible knowledge. Here, the curation author acts as a teacher, curation user is a student, and the goal of curation is knowledge transfer.

2.2 Roles in the Curation Process

Three main roles could be identified in every successful curation process. The first role is dedicated to the person who curates the content (i.e., the author). This role is responsible for selecting and filtering the content to be curated, adding the new value in form of additional content, conclusions, or insights, and targeting the users who could benefit from the curated content. The second role is dedicated to the people who use the curated content. This role represents the core users of the curation. In the case of curation for one's own purposes, the person conducting this role could coincide with the person who curates the content. The third role is dedicated to analysts. In order to improve the quality of curation, as well as its importance and reach, an efficient measurement of the successfulness of the curation is required. While automated and intelligent tools do exist for this task, human involvement is still needed, in order to fully improve the curation process. Output created by this role could be used by authors (first role) to improve the curation, thus closing the circle.

User roles in curation:

1. Author - gathers and presents content
2. User - gains knowledge or some other insight from the curated content
3. Analyst - gains insight into the information flow

2.3 Classification Method

Several different criteria for classifying content curation tools could be defined. Here, we have chosen 14 different (but not mutually independent) criteria and evaluated the top tools currently available.

1. User targeting
2. Content personalization
3. Media type
4. Adding content
5. Content structure
6. Social network integration
7. Analytics
8. Collaboration
9. Content and user credibility
10. Content suggestion
11. Re-usability
12. Access model
13. Search model
14. Content tracking (evolution checking)

User targeting is one of the first criteria to be considered, when evaluating curation tools. In different curation settings, this criterion bears different levels of importance. For instance, in a setting were content is curated for knowledge transfer (from the curation author to a user), it makes little to no sense to charge authors for curating, but in a marketing setting, charging the author is reasonable (because it enables the author to promote his/her product/idea). Charging the curation users on the other hand (or even requiring the users to register), usually depends on the scale of the audience to be engaged (e.g., if Wikipedia required paying or even registering in order to access, it wouldn't have nearly as many users as it has now). For the knowledge transfer curation setting, we propose (based on the prior experience with the Forum portal [21]) that open access is usually necessary to achieve the best impact.

User targeting criterion:

1. Free
2. Free for reading, price for authors
3. Pay for reading, free for authors
4. Everyone pays

Content personalization can be a very important criterion when considering curation tools. If the content is not personalized in any way (either by asserting an author's opinion or some other insight, or by personalizing it for the reader) the curation is merely an aggregation of the content. While content aggregation and filtering could be useful in every curation setting (marketing, information sharing, and knowledge transfer), real benefits and challenges of the curation only begin when curation author is enabled to add additional value on top of the aggregated content. Tailoring the content to the individual needs of each one of the content curation users is one of the newest challenges, and requires identifying the user's current perspective (in form of prior user knowledge, context, affinities, language, background, etc...), as well as adapting the content accordingly.

Content personalization criterion:

1. Unpersonalized
2. Author based
3. User based
4. Both user and author based

Content media type is an important criterion for classifying a curation tool. Some of the most popular curation tools only consider textual content (which is not strange, considering the fact that the majority of the content on the web is text). In the information sharing setting, textual content could be considered as a primary source for curation. This includes news and social feeds (tweets, status, etc.). In marketing and idea promoting setting, attracting the curation users usually requires more than a text; multimedia including images and videos are usual content. In the knowledge transfer setting, text is almost always necessary, but other multimedia content is often needed, too.

Media type criterion:

1. Text oriented
2. Image oriented
3. Video oriented
4. Fully multimedial

Original content adding is also one major criterion to be considered. While many tools for curation allow multimedial content to be curated, the bonding glue that ties all curated content together is often neglected, restricting the user to simple textual comments of the collected content. Enabling the user to express himself with more freedom may not be necessary in the information sharing setting, but vastly increases the possibilities in the marketing and idea promoting curation setting. In the knowledge transfer setting, allowing one to present with more expressiveness could be vital for reducing the learning curve.

Content adding criterion:

1. No content adding
2. Simple textual comments/tags/links
3. Multimedia
4. Customizable popup content (call-to actions)

Content structure is another important criterion for curation tool evaluation. When an author is curating the content, the way the curated content is structured largely influences the information flow. Content could be structured as a simple sequential stream or a list of content elements, usually enabling the user to sort the list according to some relevance. While suitable for information sharing curation setting (where naturally the newest information bears the greatest significance), in knowledge sharing setting this is usually not the case. For example, when curating some artwork or biographies (as in Austria-Forum [21]), sequential organization is not appropriate, simply because ordering is usually not easy to establish.

Content organized into collections could be more appropriate in this case. This type of organizations usually implies sorting the curated content into topics (or categories, galeries, ...). For some content types, this shallow hierarchy is usually not enough; a taxonomy is needed (in the form of a tree of nested folders, like in Pearltrees [22]). The most complicated form of organization should allow organizing the content in an arbitrary graph, thus enabling not only a system of collections and sub-collections to exist, but also for example, enabling virtual collections spanning content items across the ordinary collections.

Content structure criterion:

1. Sequential (list, stream)
2. Collection gallery (folders/topics)
3. Tree (taxonomy)
4. Graph

Integration with social networks is another important criterion. Ability to easily share curated content using social networks is almost always required when curating in information sharing or marketing and idea promoting settings. This is achieved in almost every curation tool by introducing the share/like buttons. More advanced integration allows the author to better target her/his audience by giving #hashtag or @handle suggestions for the content curated. Even higher levels of integration with social networks would provide the author with content topic or source suggestion by analyzing the part of the network the author is targeting. While this approach has little to no benefit in the knowledge transfer setting, it could assist to increase authors impact on the audience she/he needs to curate the content for. A step further would include analyzing the social network after the content has been curated, in order to allow the author to get feedback and learn about his/her audience, or even suggest the author how to improve his/her curation.

Social network integration criterion:

1. No social network integration
2. Simple share/like
3. #hashtag/@handle suggestion
4. Content suggestion (for user and audience)

2.4 Impact analytics/suggestions

The ability to use **analytics** when curating content naturally leads to the next criterion we consider. Most curation tools we analyzed lack support in this segment. The simplest form of analytics the user could gain consists of the follower or reader counts (likes). By allowing the user to tag the curation he found interesting, a simple measure of curation success is created. While simplicity of this approach is appealing for both authors and users, it carries several downsides. By giving to users only the simple like button, both sentiment of the user towards the topic and his/her opinion of the quality of the curation could get mixed up. In order to avoid this, several variables should be used instead of single one (but this comes with the price of sacrificing the mentioned simplicity of course). Also, negative sentiment can be needed when curating (for example, by analyzing which audience disliked a curation, author can also learn something during the curation process). And finally, using a higher resolution of sentiment instead of simple binary value could give much more information (e.g., using 1 to 5 stars when reviewing movies or items gives more information than a simple like/no-like). The highest level of curation analytics would include profiling the readers (by using all available data including social networks) in order to enable the author to satisfy their need for content even better.

Analytics criterion:

1. No analytics
2. Follower/reader counts
3. +/- counts
4. Reader user profiling

User collaboration is another important criterion when considering curation tools. In the simplest form of curation tool according to this criterion, an author is curating the content strictly for himself/herself, thus no sharing of any kind is enabled. When the author is curating for others, simple sharing of the curation is enabled by allowing all other users to view. More complexity is introduced when several authors are allowed to collaborate on the same curation (with rights to edit). More complex curation tools in this manner introduce roles of editors and reviewers. The best example for this would be some of the sections of Austria-Forum where several editors are enabled to curate the content, but the content gets published only after approval from the appropriate reviewers. By assigning different roles to different persons, the curation process is distributed among the group of people with different levels and areas of expertise, allowing delegating some of the time consuming work to people with lower hourly rates. This is especially important in the knowledge transfer setting, where sometimes good expert on a topic is hard to find. The highest level of user collaboration should include fine-grained access system where not only whole curations, but also the parts of curation, could be assigned to an author with some role. This approach could force content re-use, by allowing curation authors to mutually share, edit, and review parts of their curations.

Collaboration criterion:

1. No sharing (author curates for himself)
2. Simple share (single author)
3. Group of authors
4. Coarse grain editors/reviewers
5. Fine grain access system

User and content credibility could be crucial when using a curated content. The simplest curation tools allow anonymous authorship. While benefiting from simplicity, this approach severely suffers from bias. By allowing the author to remain anonymous, less pressure is put on author, likely to result in an increase of the number of contributions, but almost certainly in reduced quality (on average). While using the content curated by an anonymous single author, it is up to users to decide if the author was informed, subjective, or even honest while curating the content. One way to mitigate this is to allow other users to edit the curation in an approach commonly known as crowdsourcing. By allowing many other people besides the author himself to express their feelings and knowledge about the topic, extremely subjective or obviously false content is easily repaired. On average (provided there is a sufficient number of active users), curation made this way converges over time to a high quality one (a good example is Wikipedia). The main problem with this approach is that users usually cannot decide if the curation has already been "purified" by enough authors and whether curation has already "converged" to a good one (and, of course, fluctuations in quality could appear even after that). Other means to battle ignorant or destructive authors include self-identification. By allowing the authors to identify themselves (either by giving real biographies or by giving user names or nicknames) authors gain a certain identity in their domain. Even if someone is not using their real name, good authors usually build up a certain respect of the community over time. Their online presence in form of their earlier curations, and communications with the other users could be a good indicator of their competence. The problem with this approach is that it still suffers from the fact that ultimate decision on whether to trust the curation and its author or not lies on the user itself. In the knowledge transfer setting this is sometimes not acceptable. The ultimate (and perhaps the only one) cure is to have officially identified authors and curations. If someone is already established as an expert in a domain, his/her opinions on other curations in the same domain are informative and could provide really good insight into quality of the curation.

Content and user credibility criterion:

1. Anonymous authorship (biased)
2. Crowd-sourcing
3. Self-identified
4. Officially identified

Content suggestion is another important criterion. In most curation scenarios, the author who curates already knows what content she/he wants to curate and how to do it. Therefore, in many curation tools there are no content suggestion capabilities. However, sometimes authors need to curate the content without clear vision of the content to be curated and only knowing the topic of interest. This is often the case when author is curating for himself, or for example in information sharing setting where the author wants to keep up with the latest news in his/her domain. A simple solution is to enable users to follow all sorts of feeds (e.g., newsfeeds, tweets, Facebook statuses, etc.) and to aggregate all those sources in one place. Other, more expensive solutions, includes providing hand-picked content on topic selected by domain experts. More complicated solutions include using machine learning by introducing a recommendation engine. By analyzing the similarities between different contents related to the selected topic, the used tool is automatically providing content suggestions. A more advanced version of this approach would include various parameters that could be plugged into the recommendation engine, in order to find the best content to curate.

Content suggestion criterion:

1. No content suggestion
2. Aggregation (news feeds, follow, etc.)
3. Human selection
4. Simple recommendation engine
5. Advanced recommendation engine (tunable)

Re-usability is another important criterion. If the curating tool has an API, curations made by this tool can be easily reused in various applications. Also, when a curation tool is well integrated with the browser, curating can be done much easier.

(Re)usability criterion:

1. No API
2. Has API
3. No browser plugin
4. Has browser plugin

Access model is another important criterion. If a curation tool is primarily intended to be used for curating content for the author himself (e.g., Pocket [16]) a model where all curations are only accessible by the author himself is needed. If a curation tool is primarily intended to be used for curating the content for other users, a model where all curations are publicly accessible is needed. The purpose of these two kinds of curation tools is different, and one cannot replace the other. In a more complex scenario, the author wants to curate for others, but also needs a space where content is curated, but not yet ready to be published. In this case, it is needed for the tool to provide both types of access.

Access model criterion:

1. All private (for personal use)
2. All public
3. Both public and private

Search model is another important aspect for curation tool selection. In some types of curation tools, user searches are not enabled; this is, for example, usually the case when the tool is primarily intended for curating the content for oneself. Simple search is the most common form of searching used in curation tools and it includes searching by topic or category, author, or such. More advanced tools in this manner enable authors to add keywords and tags to curated content, and enable users to search and retrieve curated content using these tags. A more advanced approach allows the users to search not only keywords, but also structured metadata (e.g., date of contribution, related contributions, etc...), like in Serbia-Forum. The most advanced approach is also the most complicated one and includes semantic search and question understanding.

Search model:

1. No search
2. Simple search (topic/category, author...)
3. Keywording/tagging
4. Flexible metadata
5. Semantic search/question understanding

Content evolution checking criterion:

1. No checking
2. Timestamping
3. Versioning

2.5 Curation Tools Evaluation

In the following section an evaluation of several most prominent curation tools is presented based on the criteria presented in previous section

	User targeting (pricing)	Content personalization	Media type	Content adding	Content structure	Social network integration	Analytics	Collaboration	Content and user credibility	Content suggestion	Usability	Access model	Search	Content tracking
Pinterest	1	1	2	2	2	1	2	2	3	1,3	2	3	2	1
Feedly	2	2	1	1	2	1	1	0		1	2,4	3	1	1
Scoop.It	1	2	1	2,3	2	4	4	1	3	3	2,4	3	2	1
Spundge	1	2	1	2	2	1	1	2	3	1,3	2,4	3	2	1
Triberr	1	1	1	1	1	1	2	1	3	1	2	2	1	1
Bundlr	1	1	1	2	2	1	1	1	3	1,3	4	2	1	1
Instpaper	1	1	1	1	2	1	1	0	1	0	2,4	1	0	1
Pocket	1	1	4	2	1	1	1	0	1	0	2,4	1	0	1
List.ly	1	1	4	2	1	1	2	3	3	1	2,4	3	1,2	1
Storify	1	2	4	2	2	1	1	2	3	0	2,4	3	1	1
Learnist	1	1	4	3	1	1	1	1	3	0	2,4	3	1	1
Paper.li	1	2	4	3	2	1	2	2	3	1,3	1,3	3	1	1
Austria-Forum	1	4	4	3	3	1	2	3	4,2	0	1,3	3	3	3
Pearltrees	1	1	4	3	3	1	1	2	3	3	1,4	3	1	1
HeadSlinger	1	1	1	1	1	1	1	1	3	1,3	1,3	2	1	1
Bag the Web	1	2	4	3	2	1	1	1	3	0	4	3	1,2	1
Delicious	1	1	1	2	1	1	2	1	3	1,3	2,4	3	2	1
Flipboard	1	1	1	1	2	1	3	3	3	1	1,4	3	1	1
Declara	1	1	4	2	2	1	1	2	3	1,3	2,4	3	1	1

Table 1: Evaluation of different curation tools. Tools are classified using the classification system defined in the previous chapter. Legend is shown in table 2

	0	1	2	3	4
User targeting		Free	Free for reading, price for authors	Pay for reading, free for authors	Everyone pays
Content personalization		Unpersonalized	Author based	User based	Both user and author based
Media type		Text oriented	Image oriented	Video oriented	Fully multimedial
Content adding		No content adding	Comments/tags/ links	Multimedia	Customizable popup content
Content structure		Sequential (list)	Collection (gallery)	Tree (taxonomy)	Graph
Social network integration	None	Simple share/like	#hastag/@handle suggestion	Content suggestion	Impact analytics
Analytics		No analytics	Follower counts	+/- counts	Reader profiling
Collaboration	No sharing (self curation)	Simple share (single author)	Group of authors	Coarse grain	Fine grain
Content and user credibility		Anonymous	Crowdsourcing	Self-identified	Officially identified
Content suggestion	None	Aggregation	Human	Simple recommendation	Advanced recommendation
Usability		No API	REST API	No plugin	Plugin
Access model		All private	All public	Private/public	
Search model	No search	Simple search (topic/category, author...)	Keywording/tagging	Flexible metadta	Semantic search/question understanding
Content evolution checking		No checking	Timestamping	Versioning	

Table 2. Description of classification criteria used in evaluation. More comprehensive description is given in chapter 2.3.

Content tracking and verification is another important criterion. When curating a content, the author is usually collecting images, texts, videos, links, and other types of content from the web. When curating, the author is adding the snapshot of the content at some point in time. For some purposes this simplistic approach doesn't suffice. For example, when dealing with volatile content or content that is likely to be altered it is useful to somehow inform both the curation author and the curation user that the content used in curation has changed. This can be implemented using timestamping. Using this approach, both accidental and intentional corruptions could be detected, but true content cannot be recovered. Also, when some content changes, it may be useful to compare it with previous versions, thus enabling deeper insights. This could be implemented using versioning (like in Austria-Forum [21], for example).

3. CONCLUSION

First, we identified different roles in curation. Also different reasons for curation as well as curation scenarios are discussed. Several criteria for evaluation of curation tools are presented and applied on some of the most popular curation tools in use.

ACKNOWLEDGMENT

We appreciate important ideas from and discussions with Nara Kulathuramayier, Gerhard Wurzinger, and Veljko Milutinovic.

REFERENCES

[1] Ulrich Obrist, H. 2008. “A Brief History of Curating,” Manchester Cornerhouse Books. Accessed 10/06/2013.

[2] Content Curation Tools future, Curated by: Luigi Canali De Rossi.

[3] http://www.masternewmedia.org/content-curation-tools-future-part1/ Rohit Bhargava, Non Obvious: How to think different, curate ideas and Predict the Future.

[4] Stephen Rosenbaum, “Curation Nation: How to win in a world where consumers are creators,”

[5] Pariser, E,. “The filter bubble: What the Internet is hiding from you,” *Penguin UK,* 2011.

[6] Olson, P., “This Landmark Study Could Reveal How The Web Discriminates Against You,” *Forbes*, February, 2013.

[7] Changtao Zhong, “Understanding the How and Why of Online Content Curation,” Crowdresearch blog, July 15, 2013. Available: http://crowdresearch.org/blog/?p=7841,

[8] Linder, R., Snodgrass, C., & Kerne, A., „Everyday ideation: All of my ideas are on Pinterest. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems,” *ACM,* pp. 2411-2420, April 2014.

[9] http://www.pinterest.com/

[10] http://feedly.com/

[11] http://scoop.it/

[12] https://www.spundge.com/

[13] http://triberr.com/

[14] http://bundlr.com/

[15] https://www.instapaper.com/

[16] https://getpocket.com/

[17] http://list.ly/

[18] https://storify.com/

[19] http://learni.st/

[20] http://paper.li/

[21] http://www.austria-forum.org/

[22] http://www.pearltrees.com/

[23] http://headslinger.com/

[24] http://www.bagtheweb.com/

[25] http://delicious.com/

[26] https://flipboard.com/

[27] https://declara.com/

[28] http://www.europeana.eu/

[29] https://expresswriters.com/17-best-content-curation-tools-use-2015/

[30] http://blog.hubspot.com/marketing/content-curation-tools

[31] http://searchenginewatch.com/sew/how-to/2345636/26-free-or-free-to-try-content-curation-tools

Optical Character Recognition of Old Fonts – A Case Study

Pirker, Johanna and Wurzinger, Gerhard

Abstract: ***Extracting text from images of text is a challenging task. Among the greatest challenges of text recognition is the optical character recognition (OCR) of historic books. Due to low-quality images, rare fonts, and unknown dictionary standard OCR, software often fails in recognizing these texts. In this paper, we discuss existing OCR systems with a focus on learning strategies, and present an OCR model which is optimized to recognize old books. Additionally, we describe the process to measure the quality of the outcome.***

Index Terms: ***OCR, language models, text processing, string similarity, web-books***

1. *Introduction*

THE recognition and conversion from images of text have always been a challenging task for automatic data processing and information retrieval and services. In particular, the task of scanning books and making them not only digital readable, but also searchable, is important to retrieve and collect information. In this way, the information of old books, newspapers, or other texts can be a valuable and interesting source to build a strong and complete information network. Different organizations are interested in the mass scale digitization of historic documents with a focus on offering improved full-text searching [1]. Among them are efforts like austria-forum.org and serbia-forum.org in which the authors are involved. However, the optical character recognition of old books or newspapers often fails for several reasons. First, the orthography of old writings differs a lot from dictionaries available nowadays. Second, many books are not well preserved, thus, the quality of the scans is often not good. Third, the typeface, often “Fraktur” (a blackletter font), is not well covered by available OCR-software, and often lacks accuracy. Letters are written very similar (f/s), or are used in several different ways, and the distance between the characters or words is often inconsistent [2]. The most challenging part is the fact that every historic book is different and poses new challenges. Different generations of OCR systems have introduced new techniques and algorithms to enhance the recognition of books, letters, handwritings, and other written texts. Most recent systems use different learning systems, such as neural networks or export systems to enhance pattern recognition [3][4]. Such systems can achieve a very high accuracy rate of the recognition of unusual/unknown writings, such as historic fonts with well-prepared training sets. However, to optimize the outcomes of the OCR systems, different methods to improve the OCR accuracy have to be applied [1].

In this paper we introduce our OCR process for historic books and describe how the OCR accuracy can be improved in the different steps of this process. Also, we discuss how to measure and compare the quality of the results. The goal of this paper is to first analyze typical problems of OCR systems with historic texts, to present an optimized OCR process for old books, and discuss how to measure the quality of OCR techniques. We use the old book series “Wurzbach Lexikon” as a case study to describe the approach.

2. *OCR Procedure*

Different digital collections and information systems digitalize old documents, such as old books or newspapers. However, optical character recognition of old documents often poses different challenges [2]. In the following chapter we summarize the main issues.

2.1 OCR problems of Historic Books

Working with historic books, we face different problems.

1. Image problems: One of the major problems is the quality of the original book and the quality of the scan. This includes issues such as curled pages, blurred fonts, or manually edited pages (e.g. stamps or hand-written notes). Examples are illustrated in Figures 1-3. Also, the pages are often soiled, letters are hard to read

J. Pirker and Gerhard Wurzinger are with the Institute for Information Systems and Computer Media, Graz University of Technology, Austria (e-mail: {jpirekr|gwurz}@iicm.edu)

and recognize, and thus scans are often blurry and noisy.

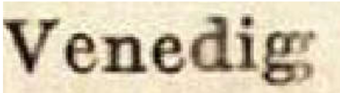


Figure 1: Blurry scan

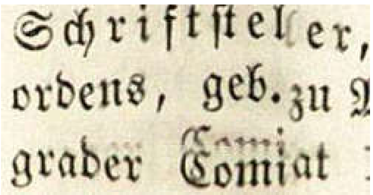


Figure 2: Curved Scans

Figure 3: Edited Original Pages

2. Font type and layout: Many historic fonts (e.g. Fraktur) are not supported by standard OCR software. Old fonts are in particular hard to recognize, because they use letters which often look very similar (see Figure 4). In earlier times, different spacing of characters was used to achieve alignment of lines. Thus, also the spacing between words and characters is often not consistent (see Figure 5). Old texts also often use different fonts in the same text (see Figure 6). Additionally, historic papers often use different and inconsistent layout structures (see Figure 7).

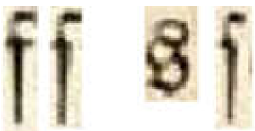


Figure 4: Different letters are often written very similarly or the same letter has different writing styles (from left to right: f, s, s, and s)

Figure 5: Different spacing of characters

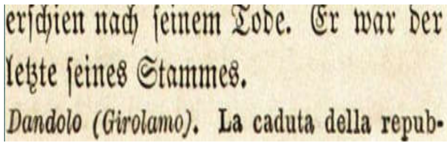
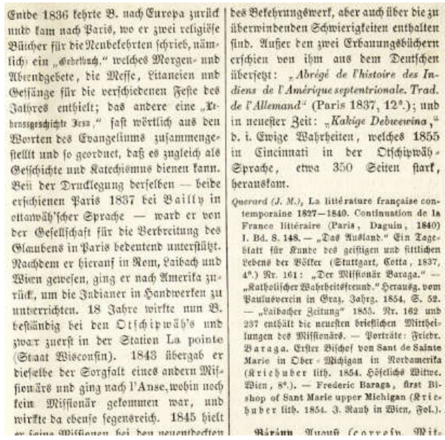

erschien nach seinem Tode. Er war der letzte seines Stammes.
Dandolo (Girolamo). La caduta della repub-

Figure 6: Different fonts and languages


Ende 1836 kehrte B. nach Europa zurück und kam nach Paris, wo er zwei religiöse Bücher für die Neubekehrten schrieb, nämlich: ein „Gebetbuch," welches Morgen- und Abendgebete, die Messe, Litaneien und Gesänge für die verschiedenen Feste des Jahres enthielt; das andere eine „Lebensgeschichte Jesu," fast wörtlich aus den Worten des Evangeliums zusammengestellt und so geordnet, daß es zugleich als Geschichte und Katechismus dienen kann. Bei der Drucklegung derselben — beide erschienen Paris 1837 bei Bailly in ottawäh'scher Sprache — ward er von der Gesellschaft für die Verbreitung des Glaubens in Paris bedeutend unterstützt. Nachdem er hierauf in Rom, Laibach und Wien gewesen, ging er nach Amerika zurück, um die Indianer in Handwerken zu unterrichten. 18 Jahre wirkte nun B. beständig bei den Otschipwäh's und zwar zuerst in der Station La pointe (Staat Wisconsin). 1843 übergab er dieselbe der Sorgfalt eines andern Missionärs und ging nach l'Anse, wohin noch kein Missionär gekommen war, und wirkte da ebenso segensreich. 1845 hielt er seine Missionen bei den neuentdeckten

des Bekehrungswerk, aber auch über die zu überwindenden Schwierigkeiten enthalten sind. Außer den zwei Erbauungsbüchern erschien von ihm aus dem Deutschen übersetzt: „*Abrégé de l'histoire des Indiens de l'Amérique septentrionale. Trad. de l'Allemand*" (Paris 1837, 12°.); und in neuester Zeit: „*Kakige Debwewina*," d. i. Ewige Wahrheiten, welches 1855 in Cincinnati in der Otschipwäh-Sprache, etwa 350 Seiten stark, herauskam.

Querard (J. M.), La littérature française contemporaine 1827—1840. Continuation de la France littéraire (Paris, Daguin, 1840) I. Bd. S. 148. — „Das Ausland." Ein Tageblatt für Kunde des geistigen und sittlichen Lebens der Völker (Stuttgart, Cotta, 1837, 4°.) Nr. 161: „Der Missionär Baraga." — „Katholischer Wahrheitsfreund." Herausg. vom Paulusverein in Graz. Jahrg. 1854, S. 52. — „Laibacher Zeitung" 1855. Nr. 162 und 237 enthält die neuesten brieflichen Mittheilungen des Missionärs. — Porträte: Friedr. Baraga. Erster Bischof von Sant de Sainte Marie in Ober-Michigan in Nordamerika (Kriehuber lith. 1854. Höfelichs Witwe. Wien, 8°.). — Frederic Baraga, first Bishop of Sant Marie upper Michigan (Kriehuber lith. 1854. J. Rauh in Wien, Fol.).

Báránn August (correfp. Mit-

Figure 7: Layout of old books

3. Missing knowledge base: Traditional OCR software uses knowledge bases based on contemporary dictionaries and grammatical structures to enhance the OCR procedure and does not provide historic dictionaries. Additionally, historic texts often do not follow specific orthographic structures and rules, thus words can be written differently in the same text.

Every historic book or newspaper is different. Thus, very specific and unique problems can occur for every project. In the next section, we take a closer look at the overall OCR process, and possibilities to improve the different steps.

2.2 OCR Process

Since every historic book presents new challenges to cope with, it is important to focus on these specific problems in the OCR process. In the following section, we describe the procedure of OCR with a focus on creating a learning / feature base for historic books, which can be used for improving machine learning algorithm. To improve the accuracy of the OCR process, different actions can be taken in every single step of the process [1]. Figure 8 gives an overview of the entire process.

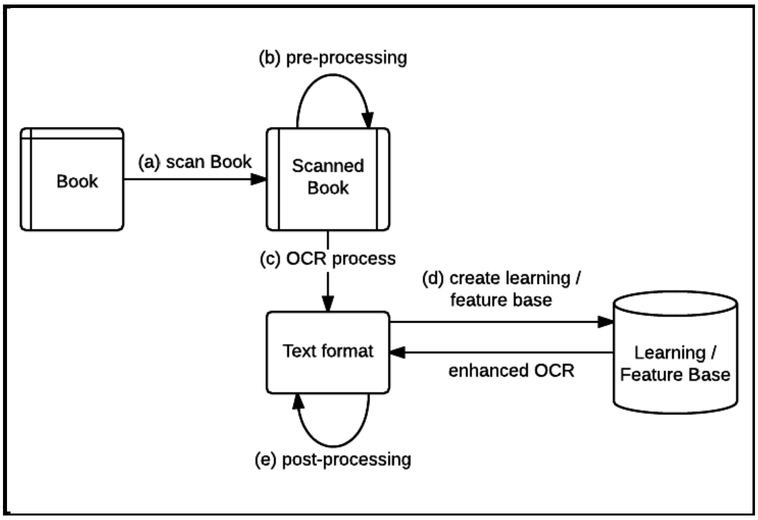


Figure 8: OCR Process

- Scanning: The first step is the scanning phase. This phase is already one of the most crucial phases. If available, scans should be made of well-preserved and clean originals. The scanning resolution should be at least 300 dpi and the output image a lossless format (e.g. tiff)
- Pre-processing: In this step, the scanned document (image file) can be manually optimized for the OCR process. This includes image editing processes such as increasing the contrast, reducing noise, or simplifying the colors.
- OCR-process: In this step, the chosen OCR system reads the images and applies an algorithm to recognize the characters. For this step, it is crucial to choose OCR software that fits the current problem and supports a training/learning algorithm. Different systems that support training are discussed in the next section.
- Create learning base: To improve the OCR-process it is crucial to create and improve the learning base for training the OCR system. This base consists of a dictionary fitting the document improved character pattern. Additionally, knowledge, such as different languages, document-specific layout, and knowledge about typical word or document patterns (e.g. topic abbreviations in an encyclopedia) can be applied to improve the OCR process.
- Post-processing: In this step, knowledge can be applied which had not yet been available to the OCR system. In a final step, the output can be corrected manually.

The last steps can be described as a feedback loop with human input for training to improve the outcomes [5]. In the next section, we discuss software packages that support training possibilities and such feedback loops.

2.3 OCR Systems

Different OCR systems use different algorithms and techniques to recognize characters. Due to the differences between historic documents, it is important to choose OCR software that supports training of the process, i.e. the user can train the software to recognize specific characters in the future. Abbyy Finereader [9] is commercial OCR software and also provides OCR dictionaries for time periods between 1800 and 1938. It has its own interface to allow the user to train. A free and open-source toolkit that provides training possibilities is Tesseract. It uses a segmentation-based training algorithm [10][11]. Segmentation-based training would use different segments of the text (e.g. characters) as an input and map it with a training process to the different character-representations. OCRopus is another open source implementation of an OCR system. In an updated version, it uses Long Short-Term Memory (LSTM) networks and line normalizations methods for an improved OCR process [12][13]. LSTM is an updated version of recurrent neural networks, a machine learning approach, which is optimized for recognition task for non-segmented applications, such as handwritings or historic fonts. Line normalization is for

example used to straighten curved words in a pre-processing step.
All these systems support feedback loops to improve the quality and accuracy of OCR. However, how can we measure the accuracy? In the next section we discuss metrics to measure the performance of OCR systems.

3. OCR Performance Metrics

To measure the performance of the OCR output, different metrics are defined in literature. Feng an Manmatha [4] describes a performance metric which used alignment of OCR output to ground truth to measures accuracy rates (ratio of the number of correctly recognized characters to total number of characters) and missing rates (ratio of the number of characters missing compared to total number of characters).
Another metric to measure similarity of two strings is the Levenshtein distance, which is defined as the minimum number of character edits to convert one string to the other with operations such as deletion, insertion, or edits [7]. To normalize this metric, we can use the following formula:

$$Accuracy_{Levenshtein} = \frac{(Length_{String1} + Length_{String2}) - LevenshteinDistance}{(Length_{String1} + Length_{String2})}$$

This formula describes the similarity of two documents (strings) based on the edit distance and relates it to the total length of the strings. The result is a number between 0 and 1, and can be used as percentage values, where 100% is a full match.

4. Case Study – Wurzbach

The book series "Biographisches Lexikon des Kaiserthums Oesterreich" is a 60-volume biographical encyclopedia collected by Constant, Ritter von Tannenberg, and Wurzbach between 1856 and 1897 with a focus on Austrian persons. Each volume consists of about 500 pages; all in all, Wurzbach is a collection of more than 24,000 biographies [14][15].

In this old collection, traditional OCR approaches mainly face the following problems:

- Typeface: The typeface used is a very old font, the scans lack quality. Additionally, the author used several different type settings.
- Missing dictionary: The languages used in this collection are mainly an old German, but also quotes, and titles in Latin, Italian, French, or Hungarian.
- Unknown words: The main part of this collection are different names for persons, titles, cities, towns and districts. Personal names are hard to verify and differ from standard dictionaries.
- Encyclopedia style: This biographical encyclopedia uses many abbreviations (e.g. letters for names of the current paragraph)

Some of these problems are very specific for this book series. However, these problems can also be used as a learning basis for building a strong feedback loop. In the case of Wurzbach, the first volume (400 pages) of the series was manually edited by the community [14]. Based on this volume, a first learning base was built containing knowledge of the personal names and a book-specific dictionary.
This edited version was also used as validation of our different learning algorithm. We used the normalized Levenshtein distance to measure and compare the accuracy of the different software systems, and to find out if applying different learning strategies would improve the accuracy.

5. Conclusion

In this paper, we tried to elucidate the process of OCR of historic books. Our analysis of this process again proves that creating a huge knowledge base of the features (dictionary, specific typeset features) is indeed the most important step of the process, as traditional OCR software most often fails to work with historic books with old fonts and inconsistent language and layout. To compare the accuracy of the OCR methods, a normalized version of the Levenshtein distance can be used. Since every historic book is different and poses its own and new challenges, the most important step of an OCR process is building a learning base. We applied this principle to the 60-volume series "Wurzbach". Thanks to a community-edited volume, we were able to build a book-specific dictionary as a learning base. Further specific “problems” of this series were used as learning “rules” to iteratively enhance the accuracy. The main contribution of this work is a model for OCR processes of historic books with old fonts. With such a model, with respect to pre- and post-processing, the accuracy of OCR of historic books can be improved significantly, compared to related approaches. Future works include a more detailed evaluation and comparison of this model.

ACKNOWLEDGMENT

This work would not be possible without the help of the involved Austria-Forum (austria-forum.org) and Serbia-Forum (http://serbia-forum.org) team. We would also like to thank the anonymous reviewers for their insightful and constructive comments. Research on Wurzbach is a project of the Austria-Forum and can be found on http://austria-forum.org/af/Wissenssammlungen/Bibliothek/Wurzbach

REFERENCES

[1] Holley, R., "How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs." D-Lib Magazine 15.3/4 (2009).

[2] “The challenges of historical materials and an overview on the technical solutions in IMPACT” [Online]. Available: https://impactocr.wordpress.com/2010/05/07/an-overview-of-technical-solutions-in-impact/

[3] Mori, S,, Ching Y. S., and Kazuhiko Y., "Historical review of OCR research and development." *Proceedings of the IEEE* 80.7 (1992): 1029-1058.

[4] Feng, S., and Manmatha, R., "A hierarchical, HMM-based automatic evaluation of OCR accuracy for a digital library of books." *Digital Libraries, 2006. JCDL'06. Proceedings of the 6th ACM/IEEE-CS Joint Conference on*. IEEE, 2006.

[5] Gupta, M. R., Jacobson, N. P., and Garcia, E. K., "OCR binarization and image pre-processing for searching historical documents." Pattern Recognition 40.2 (2007): 389-397.

[6] Stein, S. S., Argamon, S., and Frieder, O.. "The effect of OCR errors on stylistic text classification." Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006.

[7] Levenstein, V., "Binary codes capable of correcting spurious insertions and deletions of ones." Problems of Information Transmission 1.1 (1965): 8-17.

[8] Luc, V., "Google Book Search: Document Understanding on a Massive Scale." icdar. 2007.

[9] “ABBYY.” [Online]. Available: http://www.abbyy.com/recognition server

[10] Smith, R., “History of the Tesseract OCR engine: what worked and what didn’t,” in DRR XX, San Francisco, USA, Feb. 2013.

[11] Smith, R., "An overview of the Tesseract OCR engine," icdar. IEEE, 2007.

[12] Breuel, T. M., Ul/Hasan, A., Al Ayawi, M., and Shafait F., "High-performance OCR for printed English and Fraktur using LSTM networks." *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013.

[13] “OCRopus - Open Source Document Analysis and OCR system.” [Online]. Available: https://code.google.com/p/ocropus

[14] Wikimedia Foundation, “Biographisches Lexikon des Kaiserthums Oesterreich,” http://de.wikisource.org/wiki/Biographisches_Lexikon_des_Kaiserthums_Oesterreich, 08.01.2014.

[15] Wurzbach, C. V., „Biographisches Lexikon des Kaiserthums Oesterreich,“ 1891.

Johanna Pirker studied software engineering at Graz University of Technology and concluded with a thesis on virtual worlds for physics education in a joint project with Massachusetts Institute of Technology. Currently she is a researcher at the Institute of Information Systems and Computer Media, Graz University of Technology. She specializes in games and environments that engage users to learn, train, and work together through motivating tasks. Her research interests include immersive environments, game research, gamification strategies, human computer interaction, e-learning, computer science education, and information retrieval. She has authored and presented numerous publications in her field.

Gerhard Wurzinger is currently a software engineer and researcher at the Institute for Information Systems and Computer Media, Graz University of Technology. After some years of professional experience as software engineer in the automotive industry, Gerhard Wurzinger received a degree in Software Development and Business Management at Graz University of Technology. He is the technical head of the Austria-Forum.

Smart Web Services (SmartWS) – The Future of Services on the Web

Maleshkova, Maria; Philipp, Patrick; Sure-Vetter, York; and Studer, Rudi

Abstract: ***The past few years have been marked by an increased use of sensor technologies, abundant availability of mobile devices, and growing popularity of wearables, which enable a direct integration of their data as part of rich client applications. Despite the potential and added value that such aggregate applications bring, the implementations are usually custom solutions for particular use cases and do not support easy integration of further devices. To this end, the vision of the Web of Things (WoT) is to leverage Web standards in order to interconnect all types of devices and real-world objects, and thus to make them a part of the World Wide Web (WWW) and provide overall interoperability. In this context we introduce Smart Web Services (SmartWS) that not only provide remote access to resources and functionalities, by relying on standard communication protocols, but also encapsulate 'intelligence'. Smartness features can include, for instance, context-based adaptation, cognition, inference and rules that implement autonomous decision logic in order to realize services that automatically perform tasks on behalf of the users, without requiring their explicit involvement. In this paper, we present key characteristics of SmartWS, and introduce a reference implementation framework. Furthermore, we describe a specific use case for implementing SmartWS in the medical domain and specify a maturity model for determining the quality and usability of SmartWS.***

Index Terms: ***Smart Web Services, SmartWS, Semantic Web Services, Web of Things***

1. Introduction

CURRENT developments on the Web are characterized by the wider use of network-enabled devices, such as sensors, mobile phones, and *wearables* that serve as data providers or actuators, in the context of client Web applications [1]. Even though real-life objects can finally participate in Web scenarios, the use of individual and specific interaction mechanisms and data models lead to realizing isolated islands of connected devices or to custom solutions that are not reusable. Devices are increasingly network-enabled but rely on heterogeneous network communication mechanisms, use non-standardized interfaces and introduce new data schemas for each individual type of device [2]. This results in a lot of heterogeneity, in the lack of overall integration and in solutions that cannot easily be extended and reused for different application domains.

We witness similar developments in individual application areas such as in the medical, mobility and energy fields, which face these and even further difficulties. In particular, the situation is aggravated by the growing use of sensors, designated devices for monitoring and recording data, and the digitalization of domain-specific knowledge, such as recordings of trials, guidelines, common procedures, etc. These results in large data volumes [3], which are hard to integrate, are processed and managed by domain specialists, as part of their daily tasks. As a consequence, not only is it difficult to benefit from the available data in order to solve a particular problem or task, it becomes hardly possible to have an overview of all the related information or to keep up with updates.

To this end, the vision of the Web of Things (WoT) [4] is to leverage Web standards in order to interconnect all types of embedded devices (e.g., patient monitors, medical sensors, congestion monitoring devices, traffic-light controls, temperature sensors, smart meters, etc.) and real-world objects, and thus to make them a part of the World Wide Web (WWW) and provide overall interoperability. Therefore, WoT aims to build a future Web of devices that is truly open, flexible, and scalable. We aim to contribute towards achieving this goal by relying on existing and well-known Web standards and paradigms used in the programmable Web (e.g., Uniform Resource Identifiers (URIs), Representational State Transfer (REST) [5], and Hypertext Transfer Protocol (HTTP)) and employing semantic Web technologies (e.g., Resource Description Framework (RDF) [6], and Linked

Manuscript received August 15, 2015. This work was supported in part by the German Research Foundation (DFG) within projects I01, A01, R01, S01 and I04, SFB/TRR 125 "Cognition-Guided Surgery".

Maria Maleshkova (contact author), Patrick Philipp, York Sure-Vetter and Rudi Studer are with the Institute of Applied Informatics and Formal Description Methods (AIFB) and the KSRI - Karlsruhe Service Research Institute, Karlsruhe Institute of Technology (KIT), Germany (e-mail: {maria.maleshkova, patrick.philipp, york.sure-vetter, rudi.studer}@kit.edu).

Data (LD) [7]) in order to address the need for semantically integrating data coming from a variety of heterogeneous sources and for managing the ever increasing data volumes.

In order to provide a solution for addressing the challenges described above, and to be able to benefit from the WoT vision, we focus on developing Smart Web Services (SmartWS) that encapsulate 'intelligence' by implementing autonomous decision logic in order to realize or adapt services that automatically perform tasks on behalf of the user, without requiring his/her explicit involvement. SmartWS provide remote access to resources and functionalities, by relying on standard communication protocols, and also encapsulate smartness elements, as for instance:

- Context-based adaptation – automatic adjustments based on the devices' current situation;
- Cognition – for example, learning based on available data, such as previous log files, and determining optimal settings or suggesting a particular solution option;
- Inference – for example, deducing implicit knowledge based on the available data;
- Rules – for example, formal specification of common practices or established guidelines.

Therefore, SmartWS implement autonomous decision logic in order to realize services that automatically perform tasks, such as suggesting patient diagnosis, determining an optimal traveling route, or updating the temperature settings of all heaters in a house. It is only through SmartWS that we will be able to provide the added value of interoperability, scalability and integration that is needed in order to realize the WoT.

To this end, we make the following contributions:

- We provide a definition for SmartWS;
- We motivate and introduce the key characteristics of SmartWS;
- We introduce a reference implementation framework for realizing solutions based on SmartWS;
- We describe a specific use case for implementing SmartWS in the medical domain;
- We specify a maturity model for determining the quality and usability of SmartWS.

The remainder of this paper is structured as follows: In Section 2 we present the current state of the art that provides the foundation for developing SmartWS. In Section 3 we describe the main SmartWS characteristics and how we approach their development. In section 4, we provide a reference framework for realizing SmartWS and demonstrate how it can be used to support a particular use case from the medical domain in Section 5. We introduce a maturity model for classifying SmartWS in Section 6 and conclude the paper in Section 7.

With the proposed approach and the introduced framework, SmartWS can be efficiently developed and deployed.

2. *State of the Art*

Currently, there are four lines of related work that need to be considered in the context of SmartWS. These are – (i) approaches for providing remote access to functionalities and resources over the Web (Web services, Web APIs and Microservices); (ii) approaches for data interoperability and integration on top of services, (iii) approaches for encapsulating data and functionality (e.g., inference, cognition, rules) and (iv) existing approaches aiming to support Smart Services. We discuss these lines of work in the following sections.

2.1 Web Services, Web APIs and Microservices

Regarding providing remote access to functionalities and resources over the Web, the past few years have been marked by a trend towards a simpler approach for developing and exposing Web service and APIs – moving away from traditional services based on SOAP [8] and WSDL [9]. Instead of relying on the rather complex WS-* specification stack, current Web APIs rely directly on the interaction primitives provided by the HTTP protocol, with data payloads transmitted directly as part of the HTTP requests and responses. If the REST architectural principles [5] are enforced on top (also referred to as RESTful services), this provides for a more coordinated and constrained communication between the application client and the server. Furthermore, Microservices [10] represent a new emerging trend towards developing Web services by realizing a number of small, highly decoupled services that focus on doing a particular small task, thus facilitating a modular approach to system-building. As a result, complex applications can be composed of a number of small, independent reusable microservices.

These developments are taken one step further by Semantic Web Services (SWS) [11], which aim to reduce the manual effort required for manipulating Web services by enhancing services with semantics. The main idea behind this research is that tasks such as the discovery, negotiation, composition and invocation can have a higher level of automation, when services are enhanced with semantic descriptions of their properties. We use the research on SWS and develop it further to provide semantic APIs.

Instead of being simply an endpoint associated with a communication protocol, semantic APIs benefit from the Linked Data principles [7] and semantic technologies in order to provide access to all resources linked to a particular entity or to allow the retrieving of data based on the concepts and properties that is it linked to.

In terms of documenting the interfaces, we base the development of SmartWS on a lightweight implementation of the interfaces – by using APIs, and semantically describe these to not only capture what the service does and how it works, but also what resources can be manipulated via what inputs and outputs.

2.2 Service Data Interoperability and Integration

Data interoperability and integration have a long history outside the field of WoT, but the adaptation to the constraints and challenges raised by achieving such tasks in the context of dynamic, distributed and interconnected systems have received little attention [12]. For instance, the traditional Extract-Transform-Load (ETL) processes rely on data transformation pipelines to create large, centralized data warehouses, on top of which analytics can be performed. This is the approach typically employed in business intelligence [13], where the data sources are relatively static and there is a need to pull all the data in the same place. A more distributed approach is one where data interfaces are being standardized to ensure a homogeneous access to various data sources. This approach is favored in the area of Web APIs, where different data sources do not need to be pulled together, but are accessed at application level through similar interfaces [14]. The disadvantage here is that it puts higher effort on the client application, as it requires identifying, access and integrating each data source separately. This disadvantage can be overcome by conforming to shared means for communication and data exchange.

Another approach has been introduced more recently through the Linked Data principles [7], where links are established between resources of various data sources, pushing them in a common graph that can be manipulated and processed as if it were a common, unique data source. This facilitates data integration on a large scale and also supports the reuse of openly available datasets and vocabularies. SmartWS are based on realizing distributed access to resources, which are integrated through the definition of links between these resources. This approach alleviates the challenge of having to integrate each data source separately, since it benefits from the links between the resources where service compositions can be made directly by defining the data that is produced by one service and consumed by the next one.

2.3 Encapsulating Data and Functionality

SmartWS can also benefit from the notion of Research Objects [15], which was introduced in scientific disciplines to encapsulate all the necessary information for the execution of isolated scientific experiments, ensuring the reproducibility and validation of their results and preventing decay. Research Objects have been successfully used in data-intensive disciplines like Genomics and Astrophysics and were originally conceived to extend traditional publication mechanisms [16] and take scientific communities 'beyond the pdf' by aggregating essential resources related to experimental results along with publications. This includes not only the data used but also methods applied to produce and analyze that data. We adopt this approach not to encapsulate data, as is the case with research objects, but rather to include 'smartness' elements within the service interface.

2.4 Towards Smart Services

The importance of developing intelligent environments for providing services and products has already been recognized as part of the Smart Service World ("Smart Service Welt") [17] vision, where services and products offered over the Internet, can be used as the basis for developing new data- and service-based business models. In this particular context "Service Platforms (Smart Services, Architecture Layer 1)" are seen as the means for implementing, configuring and composing modular value added chains. We take this vision one step further and introduce SmartWS not only as the foundation for realizing distributed solutions but also as reusable intelligent building blocks for complex applications, thus directly contributing to realizing the vision of the Smart Service World.

Furthermore, there is some specific work in the context of approaches that aim to combine Web services/Web APIs and Linked Data for realizing service invocation or composition frameworks. In particular, these approaches focus on integrating existing data services (i.e., services that provide data) exposed through Web APIs, with Linked Data principles by having services consume and produce semantic data (i.e., RDF triples). For instance, Linked Data-Fu [18] enables the development of data-driven applications that facilitate the RESTful manipulation of read/write Linked Data resources. Linked Data-Fu provides a language for declaratively specifying interactions between web resources as well as an invocation engine that performs efficiently the described interactions with the web resources.

Another approach for automatically invoking Web APIs is presented by Taheriyan et al. [19] who offer a solution that allows domain experts to create semantic models of services with the help of an interactive web-based interface. Based on samples of Web API requests and a set of vocabularies, the system invokes the service and creates a model that captures the semantics of the inputs, outputs, and their relationships. Finally, Linked Data Fragments [20] offer a new approach towards combining Web services and Linked Data by offering a solution for publishing Linked Data via a queryable API. Instead of having a solution where services consume and produce Linked Data, the authors demonstrate how existing Linked Data datasets can be exposed via a service on the Web which allows posting queries to single or distributed data sources.

Finally, there is already some work in the context of realizing platforms that to a certain extent implement Smart Services. Lee et al. present a Smart Service Framework (SSF) [21] that focuses on supporting context adaptation by providing a design for a centralized systems that is responsible for taking the context-relevant information from mobile and service agents, processing it based on background knowledge, such as device and user profile, and adapting the offered services to better suit the current users' needs. This approach is very important since it provides a first step towards achieving some level of smartness. Unfortunately it focuses only on context adaptation and the authors do not discuss how individual devices are registered to the platform, how the communication is realized and how data heterogeneity is handled. Another approach is presented in [22] where the authors focus in particular on wearable devices in the context of Internet of Things. The solution is based on an enterprise service bus (ESB), which aims to support the integration of different hardware platforms into a single application and to introduce a service-oriented semantic middleware solution. Thus the ESB, in combination with semantic descriptions of the attached devices, serves as a bridge for interoperability and integration of the different environments. The approach reuses SWS technologies and centralities the communication over the ESB, however, it does not focus on adding 'smartness' elements to the used devices or their interfaces. Finally, Lee et al. [23] and Beltran et al. [24] present two domain-specific solutions – one centered around weather information and one designed specifically for aggregating social web data. Both solutions use WoT technologies, enhance them with semantics in order to support data integration and interoperability, and use SWS approaches. However, they stop short of extending the device and data interfaces with encapsulated intelligence.

In the following section we describe how we benefit from the current state of the art and take developments one step further in order to introduce Smart Web Services.

3. *Foundations for Smart Web Services*

SmartWS not only enable remote data access and modification, and benefit from the Linked Data principles, but also comprise 'smartness' elements in terms of context-based adaptation (e.g., changing the weather information based on the current location), cognition (e.g., learning from past experience in order to determine optimal settings), inference (e.g., automated deduction of knowledge), and rules (e.g., formal specification of heuristics or guidelines) that implement autonomous decision logic (see Figure 1).

There are a number of definitions for smart services [21]. Usually these definitions are used to describe the offerings to the users and what they can achieve by calling the service – technical or business services. Still there is no single, generally accepted definition; especially not in the context of the technical implementation of services – as Web services or APIs. Therefore, we first give our definition of SmartWS, and subsequently specify the characteristics that SmartWS should exhibit.

SmartWS *are Web APIs (conforming to standard Web technologies (HTTP, URIs)) that consume and produce semantic data (RDF) and encapsulate autonomous decision logic.*

SmartWS benefit from previous research and introduce the innovative aspect of encapsulating intelligence elements directly as part of the service. As a result, SmartWS automatically adjust configurations, adapt to context or trigger events, based on the input data, thus enabling the implementation of actuators on top of the smart APIs. SmartWS are Web APIs that consume and produce Linked Data and, in addition, encapsulate rules and inference, for instance, in order to automatically deduce further knowledge that is not stored in the dataset but can be derived based on the existing facts. They implement domain or use case-specific rules that can be used to make conclusions about the outputs, based on the provided input (for example, given a temperature of 20 degrees or more, the heaters should be turned off). They can also use the relationships between concepts and instances in order to make inferences. As such, instead of only providing access to resources or existing functionalities, SmartWS are intelligent enough to deduce additional knowledge, trigger events or directly update configurations.

3.1 Towards SmartWS

The approach followed towards realizing the vision of SmartWS is based on three main pillars – (i) semantic technologies, (ii) remote access to resources via Web services and (iii) 'smartness' elements in terms of, for instance, context-based, adaptation, cognition, inference and rules. This combination provides an innovative, up-to-date unexplored line of research that, as demonstrated by the use case implementation (see Section 5), shows promising results.

(i) **Semantic technologies** – As already discussed, we aim to take advantage of the Linked Data principles for publishing data, which through the use of standardized vocabularies, provides links between entities and an abundance of available datasets, enable data integration and the building of rich client applications.

(ii) **Remote access to resources** – The Linked Data principles are combined with established technologies for remotely accessing resources on the Web (e. g., URIs, HTTP and REST) thus developing interfaces (i.e., services) that consume and produce Linked Data – the service inputs and outputs are given in RDF, with formalized semantic meaning, uniquely identified resources via URIs, and links between these resources. The result is a framework for automated resources querying (service discovery), integration (service composition) and use (service invocation).

(iii) **'Smartness' elements** – These semantically enabled services are enriched with logic for context adaptation, cognition, inference and rules, which are implemented directly as part of the interface. Instead of having a dedicated reasoning engine, a complete machine learning approach for optimization, or a large set of rules, the goal here is to enhance services with lightweight intelligence elements that already provide added value. There are a lot of use cases, where a little intelligence within the service goes a long way (for example, an interface that provides access to a street light that takes as input the time of the day and the light intensity outside, and based on rules within the interface automatically updates the light intensity, without having to access a centralized control system first). The result is a set of intelligent reusable semantic Web-enabled interfaces that provide access to single resources and can be used to realize decentralized distributed solutions (see Section 4).

3.2 Characteristics of SmartWS

In this section we derive characteristics for SmartWS, based on the aspects of 'smartness' identified in the previous section (see Figure 1). These characteristics differentiate SmartWS from traditional Web services and Semantic Web Services (SWS).

***Characteristic 1. Encapsulating* 'smartness'** The need for intelligence within the building blocks of a system has already been raised and discussed in the context of autonomic computing [25], where computing systems can manage themselves, just by receiving high-level objectives as input. We adapt this concept and take it a step further by defining the feature of encapsulating 'smartness' as having services provide human-like intelligence such as decisions or reasoning. The goal is not to have machines emulate the processes of human thinking but rather to augment services with some limited lightweight degree of intelligence regarding particular tasks.

***Characteristic 2. Adding* automation** – In general, when it comes to supporting business processes, or individual activities, it comes down to completing a number of individual tasks. Some tasks can only be performed manually, i.e., by humans, while others can be either replaced or supported by machines. When tasks can be completed directly by using SmartWS, on behalf of the user but without his/her explicit involvement, this contributes to the overall automation of the system [26], and/or supported business process. For example, the heaters in a room can automatically be turned off when the temperature is above 20 degrees.

***Characteristic 3. Adding* autonomy** – This feature relates again to reducing the human involvement. However, in order for services to be autonomous, this requires to provide elements of self-actuation, such as self-monitoring, self-diagnosis, and self-actuating. If a service already implements functions for monitoring, diagnosis, and these are actuated in autonomous manner, it directly reduces the amount of required manual intervention. Furthermore, by adding autonomy, a service would also have higher adaptability to the evolving environment and contexts of a running system.

Characteristics 1-3 highlight the main differences between SmartWS, and SWS and Web services. However, SmartWS still build on semantic technologies (characteristics 4 and 5) and Web service communication standards (characteristics 6 and 7).

Characteristic 4. Linked Data prosumers – SmartWS are consumers and producers of Linked Data, i.e. 'prosumers'. The inputs and outputs of the services are in RDF.

Characteristic 5. Machine-interpretable descriptions – SmartWS are described in a way that enables the machine interpretation of the features of the services, including functionality, conditions for invocation, inputs, outputs and means of invocation, etc.

Characteristic 6. Conforming to standards – SmartWS are realized by following standards (i.e., follow what is accepted by the general community) for communication and data

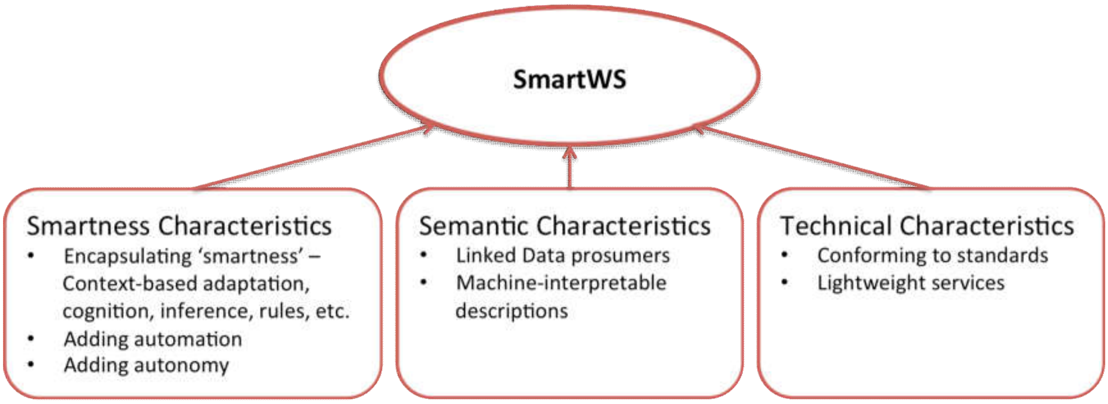


Figure 1: Characteristics of SmartWS

exchange (HTTP, URIs, XML, JSON, RDF, etc.).

Characteristic 7. Lightweight services – SmartWS are implemented as Web APIs, relying on the use of HTTP and URIs, and conforming to REST maturity level of at least 2 [27] (HTTP for message transport, XML/JSON as data format, use of resources and URIs, use of HTTP verbs).

We refer to these characteristics in more detail in order to define a maturity model for the 'smartness' of services, given in Section 6.

4. Architectural Design of a Framework Based on Smart Services

The development of SmartWS, based on the characteristics described in the previous section, can be assisted by providing guidelines, checklists and a reference architecture for implementing use case-based solutions. To this end, we present the design of a SmartWS framework, in terms of an architectural view. We provide a proof-of-concept implementation in the following section (see Section 5).

As it can be seen in Figure 2, we favor a basic three-tier solution. The bottom tier consist of all the data sources, hardware pieces, such as devices, wearables, sensors, algorithms, and software components, that should be made available as SmartWS. The middle layer consist of SmartWS that wrap the elements of the bottom layer behind a common interface, enhance it with semantics, and add further 'smartness'. The top layer represents the client side, which is facilitated via composite applications that combine the offered SmartWS into simple or complex processes.

The architectural approach that we follow is based on principles introduced by service oriented architectures (SOA) [28] and by integration systems [29, 30]. In the context of integration systems, instead of introducing wrappers and mediators, we use Web services as a way to wrap sources and realize the required

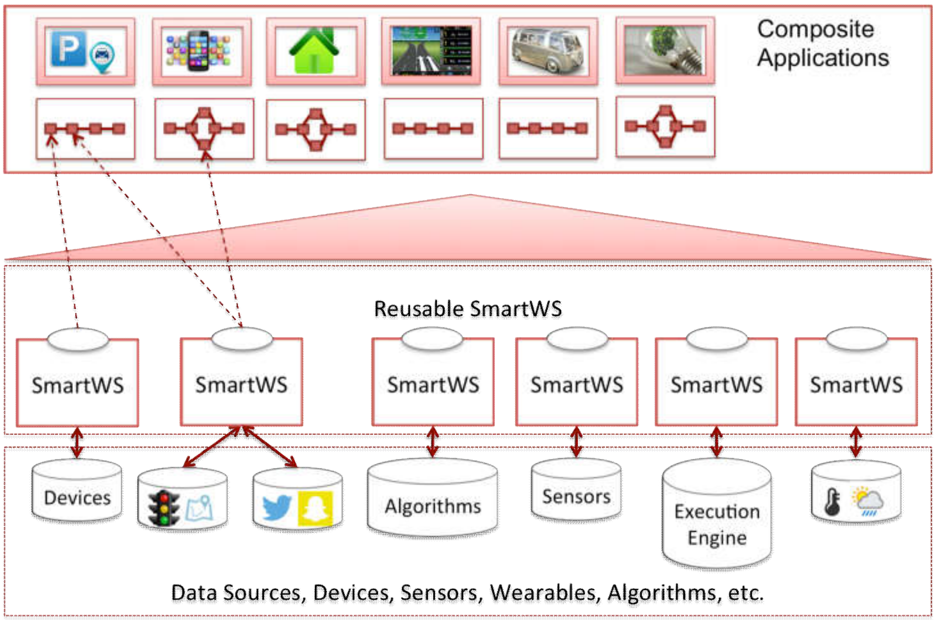


Figure 2: Reference SmartWS Architecture

mediation functionalities as part of the SmartWS implementation. Therefore, we support development approaches based on SOA and at the same time facilitate the benefits offered by an integration system. In the following, we describe each of the three layers in more detail.

4.1 Sources Layer

The sources layer consists of two main types of resources – data-centric resources and operation-centric resources. Data-centric resources are resources that are mainly related to data access and manipulation. These include, for example, repositories, databases, as well as devices and sensors that result in data streams. Therefore, we consider both static data as well as dynamic sources with a time-relevant component. Operation-centric resources focus on providing certain functionality. As examples we can have, actual Web services, algorithms, and computational software components. In addition, since any implementation can be exposed as a SmartWS in order to facilitate the development of modular distributed applications, we also advocate the approach that the actual service execution engine is made available as a SmartWS itself. The benefits are multiple, including the fact that client applications need to implement only one common interface – that of a SmartWS.

The differentiation between data-centric and operation-centric resources is not strictly required and in some instance can be difficult to determine. However, it is helpful for easing the subsequent implementation of the wrappers, in the SmartWS layer, since operation-centric sources need to be mapped to resources first in order to be able to define RESTful interfaces. While for data-centric ones this might still be required in some cases, usually the interface definition is easier.

The sources layer is extendible to comprise additional types of sources, besides the ones listed here. The only restriction is that they should be wrapable behind a SmartWS interface.

4.2 SmartWS Layer

The SmartWS layer realizes the implementation for facilitating all the characteristics described in Section 3. This starts with implementing wrappers for the individual sources, in order to be able to expose them via a RESTful interface. The thus realized interfaces are enhanced with semantics by creating a semantic service description and defining the inputs and outputs in terms of Linked Data concepts. Finally, 'smartness' elements are added, which can be simple rules (if input *temperature* > 20, directly set output to *off*, without further accessing the underlying resources) or more complex functions such as preprocessing the input that is sent to the source layer component, based on previously collected data.

The SmartWS can be registered in a centralized repository, as in the use case described in the next section, or can be stored in distributed repositories, for example, based on the specific provider.

4.3 Applications Layer

The applications layer is realized by calling individual SmartWS, creating simple compositions (sequential calling of the services) or even complex processes. The business logic that implements a workflow based on SmartWS can be encoded directly in the client application or can be controlled by another SmartWS that provides access to an execution engine. This is the layer where the actual benefit of using SmartWS becomes visible, since tasks that would normally involve some manual effort, are now completed automatically. This includes not only direct user involvement (such as, for example, turning a heater off) but also the work that needs to be invested by developers in order to implement decision logic on the client side.

In the following section we describe how we implement the SmartWS reference architecture in a specific use case in the medical domain.

5. Use Case – Realizing Smart Services as Medical Cognitive Apps

We have already tested the practical applicability of our approach and the introduced reference architecture by exploring the possibility of realizing SmartWS in the medical domain [31, 32]. In this context, we introduced a simple form of SmartWS [33, 34], which is algorithms or applications for processing medical data, accessible via a RESTful interface and consuming and producing Linked Data. Such SmartWS can provide access to formally modeled patient data in RDF, which is exposed by publishing and interlinking individual patient records by applying Linked Data principles [35]. Furthermore, SmartWS can capture medical guidelines by describing them as formalized rules in RDF or can encapsulate processing algorithms for medical imagery. The result is a distributed Web architecture for medical diagnostic systems, which support physicians while diagnosing, based on multiple SmartWS. We see this as a use case-based poof-of-concept, where a few SmartWS were implemented to test the practical applicability.

5.1 Scenario – Tumor Progression Mapping

Our work is situated within the cognition-guided surgery project SFB/Transregio125[1], which is developing assistant systems for surgeons. Here, empirical knowledge, facts and patient data are

[1] http://www.cognitionguidedsurgery.de

being combined to identify and characterize the current medical situation and its needs, and eventually perform appropriate actions. In the context of the project, we have applied our system to the Tumor Progression Mapping use case. Tumor Progression Mapping (TPM) is an approach to visualize the timely progression of brain tumors for radiologists. The process of generating a TPM produces numerous images over time, exhibiting different characteristics. Radiologists want to see the development changes of the glioblastoma since the last surgery but this requires tedious and complex tasks.

Therefore we aim to automate the workflow for tumor progression mapping, which is as follows. First, the images are stored in a centralized imagery system, and converted into a common format. A mask for the brain region is created in the next step (Brain Mask Generation, Figure 3.), ensuring that the subsequent tasks are not influenced by bones or other structures. All prevalent brain images of a patient are then spatially registered (Batched Folder Registration). The following normalization task adapts the intensities of MRI scans, thus yielding similar values for similar tissue types (Robust or Standard Normalization). If additional annotations for a patient are available, the normalization becomes more robust. Finally, the TPM can be created (Tumor Segmentation). Optional additional steps can be automatic tumor segmentation and subsequent integration into the map (Map Generation).

5.2 System Components

In this section we describe the system design for supporting the automated TPM processing pipelines. We enable medical interpretation algorithms, such as image preprocessors, that are exposed as SmartWS to automatically run when needed in potential ad-hoc workflows. This is realized by implementing the following components (see Figure 3):

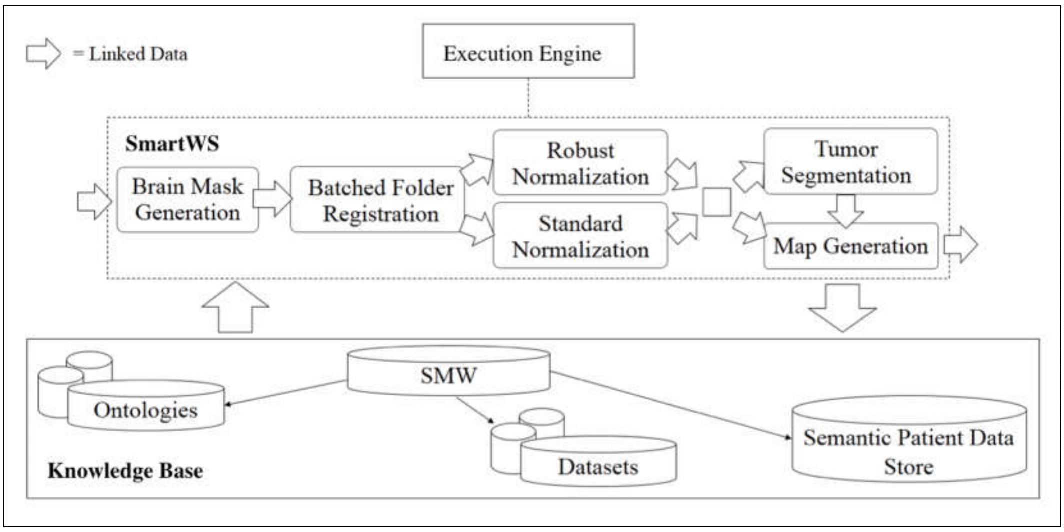


Figure 3: Brain Progression Maps Realized via SmartWS

1. A ***Knowledge Base*** containing the medical algorithms, the data to be processed as well as training and test data for supporting individual use cases. The main entry point is a Semantic MediaWiki (SMW), which is used to enable semi-structured annotations of information from both the medical and technical worlds, which is then directly available as RDF under persistent URIs. The SMW enables domain experts to semantically annotate their use cases, algorithms and data, and also to formally define their interaction. Another component of the knowledge base is the Semantic Patient Data Store. An instance of the XNAT[2] platform – an open source imaging informatics software platform – stores instance data for different medical departments and makes them accessible. As it can be seen, the knowledge base is not centralized, but includes individual databases (with data in its proprietary format – for example, XML, GIF, etc.) that are linked, annotated and accessible through the SMW. This provides for a lot of flexibility and extensibility.

2. ***SmartWS*** that provide access to medical algorithms, annotated with semantic metadata and wrapped behind a common interface, in order to be remotely accessible over HTTP.

[2] http://www.xnat.org

3. An ***Execution Engine,*** which automatically finds, initializes and runs the algorithms, based on the information stored in the knowledge base.

The newly introduced concept of SmartWS is at the core of the system. Instead of predefining or hardcoding the sequence of the used medical interpretation algorithms, we build the processing pipeline on the fly. In particular, we adopt successful Semantic Web technologies in order to create semantic annotations for algorithms, and the data they are consuming and producing. Based on these annotations we know what data is required to execute an algorithm, what this algorithm does and what are the outputs produced.

5.3 Describing SmartWS

A key element for supporting the automated composition of SmartWS is the formalized description of the algorithm properties. It consists of functional and non-functional properties.

Non-functional properties comprise the following information:

- *Name*: A unique name or identifier for the algorithm within the project.
- *Contributors*: A list of contributors to the algorithm.
- *Description*: A high-level textual description of the algorithm's functionality.
- *Evaluation Metrics*: Possible evaluation metrics for the algorithm results. These are very important, since we automatically determine which algorithm to use in the composition, based on these evaluation metrics.
- *Source Code*: Links to code repositories of the algorithm.
- *Implementation languages*: A complete list of the languages, in which the algorithm is available.
- *Service Endpoint*: This URI where the Cognitive App is executable.
- *Example Requests*: A list of URIs pointing to exemplary requests of the Cognitive App in any RDF serialization.
- *Example Responses*: A list of URIs pointing to exemplary responses of the Cognitive App in any RDF serialization.

Functional properties consist of concrete inputs and outputs of the algorithm and pre- and post-conditions of the execution:

- *Inputs: The inputs of the algorithm are either resources of type file or resources of type parameter. These resources must provide further information about their data type, about the concepts occurring in the input, about the physical format and if they are required for the execution.*
- *Preconditions: Every input must be part of a precondition to be able to specify additional constraints the algorithm has on the inputs and additional features the input should have for the algorithm to work well. Figure 4 shows the pre- and post-conditions for the Brain Mask Generation algorithm, which takes as input a headscan and two reference images (a brain atlas mask and a brain atlas image). This processing step outputs the brain image and brain mask of the headscan.*
- *Outputs: The description of the outputs has the same properties as the one of the inputs.*
- *Post-conditions: The description of the post-conditions has the same properties as the one of the preconditions. The features depict the implications on the output, in case the algorithm was executed (see Figure 4).*
- *Algorithm Class: The type of algorithm, based on a controlled taxonomy of algorithms.*

```
                  PREFIX rdf:         <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
                  PREFIX dc:          <http://purl.org/dc/elements/1.1/>
                  PREFIX sp:          <http://surgipedia.sfb 25.de/wiki/Special:URIResolver/>
?inputImage       rdf:type    sp:Category-3AHeadscan.          | ?brainImage  rdf:type    sp:Category-3ABrainImage.
?inputImage       dc:format   "image/nrrd".                    | ?brainImage  dc:format   "image/nrrd".
?brainAtlasImage  rdf:type    sp:Category-3ABrainAtlasImage.   | ?brainMask   rdf:type    sp:Category-3ABrainMask.
?brainAtlasImage  dc:format   "image/mha".                     | ?brainMask   dc:format   "image/nrrd".
?brainAtlasMask   rdf:type    sp:Category-3ABrainAtlasMask.    |
?brainAtlasMask   dc:format   "image/mha".                     |
```

Figure 4: Pre- and post-conditions for Brain Mask Generation Algorithm

The combination of pre-, post conditions, and algorithm class enable us to automatically select suitable algorithms for completing a particular task (or task sequences). Central for our ‘smartness’ aspect is stipulating evaluation metrics for an algorithm. This feature depicts if and how the system can automatically quantify results based on training samples or it can approximate them by certain variables. As a result, algorithms that are better suited for performing a certain tasks (based on the evaluation metrics) are automatically selected and favored over others.

As a result, the information needed to run a SmartWS is directly encoded in the semantic service description. The precondition of a SmartWS specifically states what data is needed to execute the algorithm and the algorithm class enables to query for specific types of algorithms. We leverage the declarative nature of the algorithm descriptions and execute the algorithms reactively on a data-driven basis (the outputs of a SmartWS are stored back in the knowledge base and used to execute all, or a predefined subset of, SmartWS that can use these outputs as inputs; the new outputs are stored again and new SmartWS can be executed, and so on). This implies that no workflows are manually defined.

5.4 Realizing TPM as Data-driven Composition of SmartWS

For the execution of the SmartWS we use the Linked Data-Fu engine [18]. It uses rules to manage and define the interaction with resources on the Web (in our case the execution of SmartWS) and to virtually integrate distributed data sources. For the knowledge base, we defined Linked Data-Fu rules to search through all ontologically annotated projects, patients and files in order to select all the patient-relevant data. The composition of individual SmartWS into a TPM processing pipeline is realized as follows – the preconditions of the SmartWS are used as IF conditions for the rules. Based on the rules, the execution engine gathers all required data and then executes all SmartWS, whose preconditions are fulfilled. The results of the executed SmartWS are stored back into the knowledge base. Therefore, the knowledgebase is enriched after every execution and this enables new SmartWS to be called. The processing pipeline can be controlled, by predefining that we want to process only a single patient, a particular set of images, only specific algorithms.

The implementation is also very flexible in terms of changing and extending the part of the knowledge based, which is being considered. Should further data sources be integrated, the execution engine has only to be enriched with more data-dependent rules. This also covers real-time scenarios, in which continuous data streams have to be polled.

6. MATURITY MODEL FOR SMARTWS

Based on the previously introduced characteristics of SmartWS and taking into account the experience gained through designing and implementing the TPM processing pipelines, we have developed a maturity model for determining the level of 'smartness' of services. The model directly relies on the SmartWS characteristics that we defined in Section 3 and captures the level of automation, autonomy and intelligent decision support that the services provide.

As shown in Figure 5, we distinguish between 3 levels of 'smartness'. Level 1 is the technical level, which ensures the good accessibility and integration in terms of using of standardized

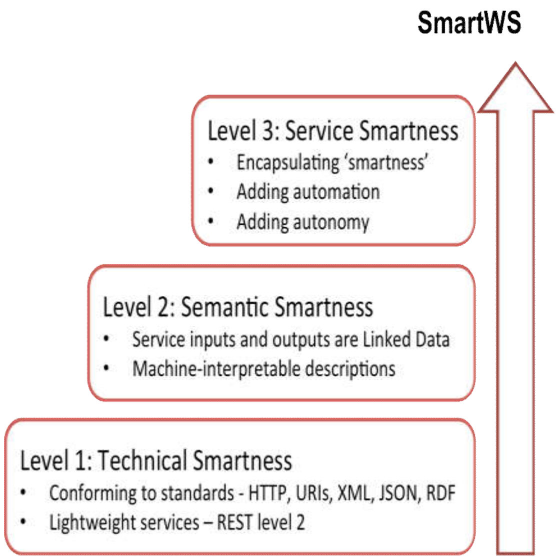


Figure 5: Maturity Model for SmartWS

access mechanisms. Level 1 of technical 'smartness' means that services are implemented as RESTful APIs, using URIs for resource identification, HTTP for communication and message transmission, as well as standard formats for data exchange, such as XML, JSON and RDF. As a result client developers know how to implement the communication with the services, solutions are more reusable and there is a direct benefit from sticking to standardized Web technologies.

Level 2 of the maturity model adds semantics in terms of both the data produced and consumed by the services as well as the actual service description. As a result services can automatically by found based on the needed functionality or the available input/expected output data. Furthermore, compositions can be made by benefiting from the Linked Data characteristics of the inputs and outputs. Similarly, already developed approaches in the context of SWS can be reused and applied. Finally, the automated execution of individual SmartWS and of compositions is supported by REST and Linked Data execution engines, such as the Lined Data-Fu engine.

Level 3 of the maturity model captures the actual added value of SmartWS. Services encapsulate 'smartness' elements (such as context adaptation, cognition, inference and reasoning), which are implemented directly as part of the service interface. Therefore, the services have their own decision logic and require less user involvement, since the SmartWS act directly without requiring further actions. For example, the heaters in a room can directly be turned off, if the temperature is above 20 degrees, without having the user do it manually. SmartWS do not only add to the level of automation but also enhance the implementation of the system. In particular, we can save on communication and data transfer between the client and the server back end, since certain outputs can be determined by the SmartWS directly based on the input (such as setting the heater to 'off') without accessing the data store or the processing component on the server. SmartWS enhance the automation and autonym of the implemented system, which are key features and a prerequisite for realizing complex use cases as envisioned by the WoT.

7. *Conclusion*

Current developments in the context of WoT call for new 'smarter' solutions that can handle the increasing complexity and heterogeneity resulting from the joined use of multiple mobile devices, sensors, wearables and data sources, in advanced client application scenarios. This situation is aggravated by the increasing data volumes that need to be handled and interpreted. To this end we introduce Smart Web Services (SmartWS), which encapsulate 'smartness' by implementing autonomous decision logic in order to realize or adapt services that automatically perform tasks on behalf of the users, without requiring their explicit involvement. SmartWS provide remote access to resources and functionalities, by relying on standard communication protocols, and are enhanced with semantics in terms of the inputs and outputs as well as the actual service description.

In this paper we present the key characteristics of SmartWS, and introduce a reference implementation framework for realizing systems based on SmartWS. We provide a proof-of-concept implementation of SmartWS in the medical domain and specify a maturity model for determining the quality and usability of SmartWS. It is only through SmartWS that we will be able to provide the added value of interoperability, scalability and integration that is needed in order to realize the WoT.

Acknowledgment

We would like thank Michael Goetz, Christian Weber, Marco Nolden, Klaus Maier-Hein from the German Cancer Research Center, Heidelberg, Germany for their contribution to the original development of the BTM use case published in [33, 34].

References

[1] D. Guinard, V. Trifa, F. Mattern and E. Wilde, E. "From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices. In Architecting the Internet of Things," *Springer*, Berlin Heidelberg, 2011.

[2] Wood, Alex. "The internet of things is revolutionizing our lives, but standards are a must," theguardian.com. *The Guardian*. Retrieved 31 March 2015.

[3] Violino, Bob. "The 'Internet of things' will mean really, really big data," *InfoWorld*. Retrieved 9 July 2014.

[4] D. Guinard, V. Trifa, V. and E. Wilde. "A Resource Oriented Architecture for the Web of Things," *Proceedings of the Internet of Things Conference*, 2010.

[5] R. T. Fielding. "Architectural styles and the design of network-based software architectures." PhD thesis, University of California, 2000.

[6] G. Klyne and J. J. Carroll. "Resource Description Framework (RDF): Concepts and Abstract Syntax." *W3C Recommendation*, 2004.

[7] T. Heath and C. Bizer. "Linked data: Evolving the web into a global data space." *Synthesis lectures on the semantic web: theory and technology 1*, no. 1 (2011): 1-136.

[8] N. Mitra. "SOAP Version 1.2 Part 0: Primer." *W3C XML Protocol Working Group* (2002), Working Draft, 2002.

[9] R. Chinnici, H. Haas, A. A. Lewis, J.-J, Moreau, Orchard, D. Sanjiva, and W. Sanjiva. "Web Services Description Language (WSDL)," Version 2.0 Part 2: *Adjuncts. World Wide Web Consortium, Recommendation* REC-wsdl20-adjuncts-20070626, 2007.

[10] J. Thones. "Microservices." *IEEE Software* 32, no. 1 (2015): 116, 2015.

[11] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, and J. Domigue. "Enabling Semantic Web Services." *The Web Service Modeling Ontology. Springer*, Berlin, Heidelberg, 2006.

[12] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffle. "Vision and challenges for realising the Internet of Things," *Publications Office of the European Union*, 2010.

[13] E. Vitt, M. Luckevich, S. Misner, G. Beijering, and B. van Spijker. "Business intelligence," Vol. 23. *Microsoft Press*, 2002.

[14] A. Gustavo, F. Casati, H. Kuno, and V. Machiraju. "Web services." *Springer* Berlin Heidelberg, 2004.

[15] K. Belhajjame, O. Corcho, D. Garijo, J. Zhao, P. Missier, DR. Newman, R. Palma, S. Bechhofer, E. Garcia-Cuesta, JM. Gomez-Perez, et al. "Workflow-Centric Research Objects: A First Class Citizen in the Scholarly Discourse," *In proceedings of the ESWC2012 workshop* SePublica2012, May 2012.

[16] S. Bechhofer, I. Buchan, D. De Roure, P. Missier, J. Ainsworth, J. Bhagat, P. Couch, D. Cruickshank, M. Delderfield, I. Dunlop, M. Gamble, D. Michaelides, S. Owen, D. Newman, S. Sufi, and C. Goble. "Why linked data is not enough for scientists," *Future Generation Computer Systems*, 2011.

[17] acatech, "Smart Service Welt – Umsetzungsempfehlungen für das Zukunftsprojekt Internetbasierte Dienste für die Wirtschaft," *Deutsche Akademie der Technikwissen-schaften.* Abschlussbericht, March, 2015.

[18] S. Stadtmüller, S. Speiser, A. Harth and R. Studer. Data-Fu: "A Language and an Interpreter for Interaction with Read/Write Linked Data," *Proceedings of the 22nd international conference on World Wide Web* (WWW), 2013.

[19] M. Taheriyan, C.A. Knoblock, P. Szekely, and J.L. Ambite. "Rapidly integrating services into the linked data cloud," *Proceedings of the 11th International Semantic Web Conference* (ISWC), 2012.

[20] L. Rietveld, R. Verborgh, w. Beek, M. V. Sande, S. Schlobach. "Linked Data-asa-Service: The Semantic Web Redeployed," *Proceedings of the European Semantic Web Conference* (ESWC), 2015.

[21] J. Y. Lee, M. K. Kim, H. J. La, and S. D. Kim. "A software framework for enabling smart services," *Paper presented at the meeting of the SOCA*, 2012.

[22] P. Castillejo, J.-F. Martínez, L. López and G. Rubio. "An Internet of Things Approach for Managing Smart Services," *Provided by Wearable Devices*. IJDSN 2013, 2013.

[23] J. W. Lee, Y. W. Ki, and S. Kwon. WISE: "An Applying of Semantic IoT Platform for Weather Information Service Engine," *Paper presented at the International Semantic Web Conference (Industry Track)*, 2014.

[24] V. Beltran, A. M. Ortiz, D. Hussein, and N. Crespi. "A semantic service creation platform for Social IoT," *Paper presented at the meeting of the WF-IoT*, 2014.

[25] Jeffrey O. Kephart and David M. Chess. "The Vision of Autonomic Computing," *Computer*, 36(1):41–50, 2003.

[26] Jammes, F., Smit, H.: "Service-oriented paradigms in industrial automation," *IEEE Trans. Industr. Inf.* 1(1), 62–70, 2005.

[27] M. Fowler. "Richardson Maturity Model: Steps toward the glory of REST," 2010.

[28] A. Rotem-Gal-O. "SOA Patterns," Shelter Island, NY: *Manning Pubications Co.*, 2012.

[29] G. Wiederhold. "Mediators in the Architecture of Future Information Systems," *IEEE Computer* 25 , no. 3 (1992): 38-49, 1992.

[30] S. Suwanmanee, D. Benslimane, P.-A.Champin, and P. Thiran. "Wrapping and Integrating Heterogeneous Relational Data with OWL," *Paper presented at the meeting of the ICEIS* (1), 2005.

[31] T. Weller and M. Maleshkova. "A RESTful Approach for Developing Medical Diagnostic Supporting Systems. Proceedings of Services and Applications over Linked APIs and Data," *European Semantic Web Conference (ESWC)*, 2015.

[32] P. Philipp, M. Maleshkova and A. Rettinger. "A Semantic Framework for Sequential Decision," *Proceedings of International Conference on Web Engineering (ICWE)*, 2015.

[33] P. Gemmeke, P. Philipp, M. Maleshkova, M. Goetz, C. Weber and B. Kaempgen. "Composing Pre-processing Pipelines for Medical Images Using Web APIs and Semantics," *In Consuming Linked Data (COLD)*, International Semantic Web Conference (ISWC), 2014.

[34] P. Philipp, D. Katic, M. Maleshkova, M. Goetz, C. Weber, A. Rettinger, S. Speidel, B. Kaempgen, S. Zelzer, A. Wekerle, R. Studer, R. Dillmann, H. Kenngott, B. Müller. "Towards Cognitive Pipelines of Medical Assistance Algorithms," *Proceedings of Computer-Assisted Radiology and Surgery (CARS)*, 2015.

[35] K. Maerz, M. Hafezi, T. Weller, M. Nolden, S. Zelzer, M. Maleshkova, M. Wagner, S. Engelhardt, H. Kenngott, A. Rettinger, R. Studer, A. Mehrabi and L. Maier-Hein: "Towards Knowledge-based Liver Surgery - Holistic information processing for surgical decision support," *Proceedings of Information Processing in Computer-Assisted Interventions (IPCAI)*. 2015.

Digitalization of Scientific and Cultural Heritage in Dissemination of Innovative Teaching Models

Lajbenšperger, Milica; Šegan, Marija; Vujošević, Sandra; and Ognjanović, Zoran

Abstract: ***This paper presents a new cycle of Mathematical Institute of the Serbian Academy of Sciences and Arts program "Digitization of cultural and scientific heritage with the application in high school and university teaching of mathematics, computer science, astronomy, history and Serbian language" (2011-2014, 2015-2017). Its goal is digital literacy of teachers and students through the implementation of principles and digitization tools in educational work of high schools. Unlike the past work with big student groups, the authors are now testing the hypothesis of small homogeneous group efficacy. This paper's hypothesis is that the students' success depends on individual differences, as well as Blum's research findings that talk about variations in school learning and success depending on students' initial cognitive and affective characteristics, and the quality of the lectures. The first results show that the work in small homogeneous groups has motivated the teachers involved in the program to implement more of the innovative teaching models, which in return resulted in a better students' success. After the evaluation of this year's project, the authors will try to asses if this kind of work organization will be more efficient than the present-day organization.***

Key words: ***digitization of cultural heritage, innovative teaching models, homogenous groups.***

1. INTRODUCTION

THE main characteristic of the modern society is a constant need for fast improvement, specialization and development in all the fields. However, the field of formal education, that is, the schooling system, represents a slow system whose development and tackling of all the challenges that the modern society sets, often depends only on enthusiastic efforts of the individuals who introduce novelties in their work.

Mathematical Institute of the Serbian Academy of Sciences and Arts, in cooperation with the Center for the Promotion of Science from Belgrade and School for Mechanical and Electrical Engineering "Goša" from Smederevska Palanka, has started in 2011, a program titled "Digitization of cultural and scientific heritage with application in high school and university teaching of mathematics, computer science, astronomy, history and Serbian language". The program is a part of a project supported by the Ministry of Education, Science and Technological Development, iii044006: „Development of new information and communication technologies, through the use of advanced mathematical methods, with application in medicine, telecommunications, energetic, protection of national heritage and education". The program's goal is the popularization of science, technology and culture among the youth from the underdeveloped municipalities of Republic of Serbia, through the process of connecting the digitization of scientific and cultural heritage, and educational work of high schools.

Starting hypothesis of the program was that the principles and technology of heritage digitization (since they are based upon various theoretical and practical knowledge crossovers), can be efficiently used in the history, geography, computer science, mathematics, and linguistics lectures, especially in the schools that are situated far away from big city centers. Its multidisciplinary aspect is recognized as a motivational factor for teachers to develop their creativity, and apply the innovative models of lectures more widely. As for students, it allows further specialization in the field of science, technology and culture.

Manuscript received October 16, 2015. Milica Labenšperger, Centre for Education 'Playground of Imagination', info@igralistemaste.com; Marija Šegan, Mathematical Institute SASA, msegan@mi.sanu.ac.rs; Sandra Vujošević, Mathematical Institute SASA, sandrav@mi.sanu.ac.rs, Zoran Ognjanović, Mathematical Institute SASA, zorano@mi.sanu.ac.rs. Contact person: Marija Šegan, Mathematical Institute SASA, msegan@mi.sanu.ac.rs.

The first results of the program showed that the suggested model enables:

1. Teachers' and students' development of transversal competences (knowledge and skills), through the use of innovative disciplines and technologies.
2. Encouraging motivation for studying educational curricula in regular schooling, through proactive participation in the programs based on inter-curricula connected content.
3. Increase of the school's competitiveness, through cooperation with experts and openness towards the local community and relevant establishments and institutions.

The Technical and Civil Engineering School "Neimar" from Niš joined the program in 2015.

2. Theoretical Frameworks and the Work Methodology

The innovations in the educational process are conditioned by many factors, amongst which, in the last decades, we recognize and often highlight two major groups: 1) the development of information and communication technologies that change current and introduce new methods and forms of teaching, and organization of the teaching process, and 2) the individual characteristics of individuals participating in the education processes. Even though these two groups of factors come from different sources, they have a strong interaction and they complement each other.

Fast development and progress of information and communication technologies, along with constant upgrading of educational technologies; cause the changes in methods and types of lectures and their organization. The new models of work, like the programmed lectures, computer science lectures, and distant learning, explicitly request the use of computers. On the other hand, for example, the interactive, project-orientated, and integrative lectures require substantially more active and responsible role of participants in the process of studying and teaching, than it is the case with the traditional education models. To some of the participants in the educational process it is easier, and to others it is more difficult to take on these roles. That fact implies the existence of individual differences and the significance of timely recognition, understanding, and maximum engagement of individual teachers' and/or students' potentials.

2.1 Individualized and Differentiated Lectures

Although the idea about individualized lectures isn't a new one [2],[4],[11], it presents a „permanent didactical innovation" [5]. The essence of the individualized lectures is in the organization of the work itself that is coordinated with the individual differences among the students – from the differences in mental abilities, motivation, interests, attitudes, to the learning styles etc. [12]. As the school's system doesn't function on the "one-on-one" principle, the individualization is acquired by differentiation, that is, by creating different conditions in which the lectures are held in. That could be accomplished by classifying students with similar individual characteristics into homogeneous groups, or by distributing the content and the tasks inside of one heterogeneous group (for example, giving students to work on different aspects of the same theme).

2.2 Innovative Teaching Models as Challenges to Modern Pedagogy

The innovative teaching models present challenges to modern organization and conducting of the educational process by changing the role of a teacher. The teacher is not only the lecturer, but the organizer of the lectures, and a partner in immediate communication among students and the teaching content. Also, a student leaves his or hers passive role and takes on the initiative and the responsibility for his or hers own studying. Therefore, apart from the activity, the motivation and the interest [8] are present among the students and the teachers. It is interesting to find that the preparations for the work on the digitization project motivated the teachers involved in the program to apply the innovative models in everyday lectures even more than before.

"The integrative teaching is the innovative model in which there are no strict boundaries between certain subjects, but the educational units of different fields are studied integrally, in their inter-curricula connection" [3]. With its multidisciplinary approach, it is the project of scientific and cultural heritage digitization that succeeds in connecting teaching contents from different curricula, and that directs the participants in the process of acquiring and applying the knowledge towards further specialization in the field of science, technology, and culture.

Also, unlike the traditional pedagogy that is orientated to knowledge and reproduction, the innovative pedagogy is concentrated to abilities and development of students' personalities [8].

Another innovative model that is significant for advancing the abilities, development of personality, and social competency, is the interactive teaching. „It is only by the active students' work that we provide effective learning. Apart from that, the interaction is a prerequisite for development of the interpersonal qualities of students' personalities“[3:p90]. The same author cites that „this learning is based on the achievement, positive connections among individuals, and achieving social competence in the process of psychological adjustment“. The facts that go in favor of it are that the project of heritage digitization is continued due to the request of the students themselves. Also the participants' network is getting wider and the students-participants in the project initiate activities themselves and suggest ideas for new teaching contents.

2.3 Homogeneous Groups

Unlike the previous work with big groups of students who were applying for certain workshops based only on their own interests, the idea of this year’s project activities was to test the hypothesis on efficiency of small homogeneous groups. Namely, the authors’ assumption is that the students’ success depends on individual differences. Also there are Blum’s research results [2] that talk about variations in school learning and successes depending on the starting cognitive (foreknowledge, reading with understanding, intelligence), and students' affective characteristics (affinity towards the subject, affinity towards the school, self-image), and the quality of the lectures (instructions and explanations, students' active participation, reinforcement, feedback and corrective procedures). Therefore, the work process of this year's program was conducted according to the stages of the individualized lectures [4:p51-52]. The first stage was the forming of the homogeneous groups.

Using a questionnaire made by the authors, the starting students' characteristics were registered, and, based on the received data, small homogenous groups were formed. Further on, each group was given special tasks, instructions and explanations, and the guidelines for their independent work. For example, one of the activities was a joint tour of the National Museum. However, the students were given different tasks according to the starting characteristics of their small homogeneous groups. Also, after every conducted task, the students were given feedback on the quality of their work, and some of the results were presented on the Facebook page of this year's program.

3. *Review of This Year's First Results*

The data is still being collected, and their interpretation is still to be conducted. Considering the basic author's hypothesis, at this moment we can say that: in all of the planned activities, the students received tasks with instructions and guidelines for work in small groups (3-6 members), as well as the feedback information about the success of the realized tasks. However, due to the dispersal of the participants, the homogeneity of the groups by predicted parameters (students' interests, students' starting cognitive and affective characteristics) wasn't complied through all the phases of the program. Also, while carrying on with the different tasks, the individual members crossed from one group to another, therefore, the success and accomplishment of small groups is also inconclusive. This doesn't undermine the value of the program as a whole because the selected program's participants (23 participants that went through all the phases of the program), considering their basic characteristics, should be treated as a homogeneous group in regard to the entire population of the school that was included in the project.

Some examples of this year's program's results:

1. The teachers-participants on the project have readily accepted the roles which the innovative teaching models required of them. The active involvement of teachers in the process of selection of cultural objects shows their motivation, accountability and collaborability. For example, the goal of the teacher from the Technical and Civil Engineering School “Neimar” from Nis was to familiarize the students with the local heritage, without endangering the students’ safety. Hence, he consulted the representatives of the Institute for the Protection of Cultural Heritage from Nis to determine the monuments’ availability and accessibility before making his final choices[13]. This way, not only did he take an active role and responsibility for the realization of the project’s activities, but he also made an unofficial cooperation (which is, unfortunately, usually hard to make) between the educational, research and cultural institutions.

2. In the project's preparation phase, the teaching material was written in a language familiar to the students-participants of the program [14].

3. The prepared tasks for small groups were made by a multidisciplinary team (teachers and experts from various fields). That is how each expert tour represented a small project within the whole program that directly supported interactivity, as well as inter-curricula connection, that is integration. Also, tasks or "mini projects" were made in a way that inspired students to take initiative and responsibility for their own studying [15].

We will be able to talk more about the connection between the students' basic characteristics and individual achievements after detailed analysis of all the gathered data. General impressions and the first gathered data favor the quality of the realized program. So far, through the analysis of students' success after finishing the project, there has been noticed the increase of interest in the local history, visiting the cultural and historical monuments, research work, etc.

Based on students' history marks, these are the indicators:

Mark	Before the project	After the project
5	4	7
4	3	1
3	1	/
2	/	/
1	/	/
Total	8	8

The average mark for history among 8 students who were involved in the project has risen from 4.38 to 4.88.

These indicators justify the goals of the project and present an additional stimulus for further work and continuation of collaboration.

4. Conclusion

The importance of individualized and differentiated lectures is reflected in encouraging the individual work, forming the positive motivation for learning, liberating individual student's potential abilities, and contributing to development of creative thinking. The other mentioned innovative models have the same great pedagogical and educational potential for successful work with students with different starting individual characteristics.

Heritage digitization project, with its numerous results in the past five years, accentuates the importance and encourages the use of different innovative models of lectures. Will the organization of work in small homogeneous groups be more efficient than the previous work in bigger heterogeneous groups; we'll try to asses after the evaluation of this year's project.

References

[1] Altaras, A.D., (2012), IOP for talented pupils – a selection of examples for the module, Annex 7: Basic educational solutions for talented pupils (supporting tools), *available at*:
http://www.cipcentar.org/i_roditelji_se_pitaju/PDF/Drustvena_i_obrazovna_ukljucenost/IOP%20za%20darovitog%20ucenika.pdf (accessed: March 03, 2015).

[2] Bloom, B.S., "Human characteristics and school learning," USA: McGraw-Hill book company, 1976.

[3] Đorđević, V., "Inovativni modeli nastave," Obrazovna tehnologija, br. 4/2007, 76-97, 2007.

[4] Lazarević, V., "Individualizovana nastava," Obrazovna tehnologija, br. 2/2005, 47-60.

[5] Marković, M., "Individualizovana nastava," Obrazovna tehnologija, br. 2/2005, 61-662005.

[6] "Project DIGITARIJUM," DIGITARIJUM Homepage, accessed October 15, 2015, available at http://digitarijum.rs/project/about-project/.

[7] Radovanović, I., Radović, V., i Tadić, A., (2009), „Inovacije u nastavi – bibliografija," (1983-2008), Beograd: Učiteljski fakultet, available at http://www.uf.bg.ac.rs/uf.bg.ac.rs/inovacije/dld/Bibliografija.pdf (accessed: March 22, 2015),

[8] Stepanovna, A.S. „O inovacionim i tradicionalnim modelima nastavnog procesa," Nastava i vaspitanje, br. 2/2006.

[9] Šegan et al. (2014), "Digitalizacija naučnog i kulturnog nasleđa u srednjoškolskom obrazovanju," Homepage of Ministarstvo prosvete, nauke i tehnološkog razvoja, accessed April 04, 2015, available at: http://www.mpn.gov.rs/images/content/dokumenta/obrazovne_pol/digitalizacija_naucnog_i_kulturnog_nasledja.pdf.

[10] Vilotijević, M. "Od tradicionalne ka informacionoj didaktici," Beograd:, Pedagoško društvo Srbije, 1999.

[11] Vilotijević, M., "Didaktika," Beograd: Naučna knjiga, 2000.

[12] Vulfolk, A. et al. Psihologija u obrazovanju I, Beograd: Clio, 2014.

[13] '"Example of the cultural heritage object selection in Nis," *DIGITARIJUM Homepage*, accessed November 22, 2015, available at http://digitarijum.rs/wp-content/uploads/2015/11/misanu_primer-izbora-ok_eng.pdf.

[14] Vujošević, S., "Moravski stil: Manastir Veluće," DIGITARIJUM Homepage, accessed November 22, 2015, available at http://digitarijum.rs/wp-content/uploads/2015/09/misanu_primer-nast.pdf.

[15] "Digitarijum workshops and field trip," DIGITARIJUM Homepage, accessed November 22, 2015, available at http://digitarijum.rs/%D0%B0ctivities/training/#workshops.

A Dataflow Machine Architecture for Static Dataflow Program Graphs

Verdoscia, Lorenzo

***Abstract*—Reconfigurability and huge density of today's devices constitute an ideal tool to experiment and implement new forms of computation. The number of cores integrated onto a single die is expected to climb steadily in the foreseeable future. This transition to many-core chips (thousand of cores per processor) is driven by a need to optimize performance per watt. Thus, several computer research groups are trying to understand how best connect these cores, to design parallel programs, and how many core architectures and their software can scale to the thousands of cores that hardware will be able to support in a decade. In this scenario, our question is whether it is time to revaluate dataflow and functional paradigms. Here we present a dataflow machine prototype that includes identical processors which constitute the reconfigurable environment devoted to accelerate the execution of chunks of dataflow program graphs directly into hardware. The processor consists of a set of identically thin computing units, co-designed between the homogeneous High-Level Dataflow System model and the Backus functional programming style, and a custom reconfigurable interconnect. As a result of the co-design approach, the one-to-one mapping between actors of the model and computing units of the processor happens in a straightforward manner. In this way, it is possible the execution of static dataflow program graphs without using memory to store partial results when data tokens flow from a computing unit to another; and without generating control tokens during the computation so that graph executions occurs in a completely asynchronous manner. Finally, by means of an FPGA-based demonstrator, specifically realized to validate the basic design choices of the prototype, some experimental results in solving a linear equation system with the Jacobi and Gauss-Seidel iterative algorithms are reported and discussed**

***Index Terms*—dataflow; asynchronous computing; high performance; functional programming; reconfigurable system; parallel architecture; embedded system.**

I. Introduction

THERE is still a slight inclination on the part of the HPC community to embrace the dataflow ideas in order to speed up the execution of scientific applications. The reasons are mostly of a pragmatic nature rather than technical. A dominant reason why the HPC community and, in particular, the applications programmers do not pay more attention to the advanced dataflow architecture ideas is due by the fact that, in the past, very high performance dataflow systems of commercial grade were not readily available on the market. Simulations and relatively low speed, low density prototypes of academic, and inefficiency, in terms of performance, did not made dataflow architectures attractive to computational scientists [1] [2] because they did not offer the opportunity to application programmers to run their problems faster than before. However, despite general scepticism for past disappointing results, it is coming out that dataflow systems are still a valid manner to increase performance [3]. These systems, employing FPGA (readily available on the market, nowadays) to implement dataflow accelerators, outperform most of the TOP 500 supercomputers not being paradoxically included in the list. This happens because the (re)configurable computing paradigm offers a performance of custom hardware and flexibility of a conventional processor [4] [5]. Because of this flexibility and the Intellectual Property availability, the (re)configurable approach does not only significantly accelerate a variety of applications [6] but constitutes also a valid execution platform to form programmable high-performance general-purpose systems [7]. In particular, given its fine grain nature, the static dataflow execution model is promising when applied to this platform [8] [9] [10].

In the emerging exascale computing era of expected architectures characterized by hundreds of thousands of processors with thousands of cores per chip at a 14-nm technology or less, the design of new computer systems is going to deal with challenges such as processor architecture, system power, memory bandwidth and capacity, clock frequencies, cost of data movement, programming model, reliability and resiliency [11], and prevention of dark silicon problems [12]. However, this requires as well a critical re-examination of "consolidated wisdoms" in terms of programming styles and execution models.

In the scientific and industrial community, there exists a reasonably definitive stand that envisions processors with thousands of small and simple cores [13]. They are the most efficient structures for parallel codes and provide the best tradeoff between energy consumption, performance, and manufacturability. New applications for big data and cognitive computing are more and more characterized by data-driven computation rather than conventional computation [14]. On the other hand, if thousands of core per chip are destined to be the way forward, a new parallel computing environment must be developed [15] for them. Another issue is the type of hardware building blocks (cores, networks on chip, etc.) that should be used for these systems based on many-core processors [16]. However, how to organize, design, and program them is still unclear.

To exploit the parallelism that this profusion of cores offers and to respond to data-driven necessity, our question is whether it is time to seriously revaluate the functional programming style and the dataflow paradigm, given their very strong relationship in terms of explosion, management, and execution of all the parallelism that applications present. In fact, if one hand the functional paradigm allows the natural

Lorenzo Verdoscia is with the Institute for High Performance Computing and Networking of Napoli, National Research Council, Italy (email: lorenzo.verdoscia@na.icar.cnr.it).

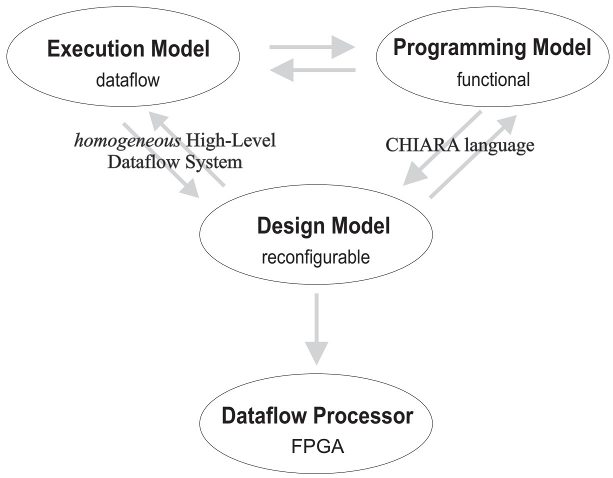


Fig. 1: Dataflow Processor design methodology.

creation of dataflow program graphs, on the other such graphs can be directly mapped and executed by a dataflow processor composed of thousands of thin, low power consuming, and identical computing units connected by means of an ad hoc reconfigurable network.

Even though dataflow is not new, general-purpose reconfigurable dataflow processors are a rather new line of research, where only few issues have been addressed yet. In this paper we first overview conceptual and practical aspects of dataflow actors with homogeneous I/O conditions, then, we concentrate on the architectural aspects of a reconfigurable dataflow processor.

The main contribution of this paper is the design and evaluation of dataflow machine architecture whose processors are constituted by a set of asynchronous, simple, and identical Computing Units (CU). Processors are dataflow program graph executors whose low-level programming language is the functionally complete subset of a functional programming language. Each executor executes static dataflow graphs without using memory to store partial results when tokens flow from a CU to another and without using control tokens but doing it in a completely asynchronous manner.

The remainder of this paper is structured as follows: Section II explains the reasons of the followed design direction for the machine architecture; Section III gives a survey of related works; Section IV presents the architecture of the dataflow machine and its demonstrator; Section V shows the performance on the machine demonstrator comparing the Gauss-Siedel with Jacobi iterative algorithms; finally, Section VI reports our concluding remarks.

II. Preamble and design directions

A. *Dataflow overview*

In the model of computation based on the dataflow graph, a program is described as a set of operator nodes, called *actors*, interconnected by a set of data-carrying arcs, called *links*. Data is passed through this graph in packets called *tokens*. Within this model of computation, the graph can be static – only one token at a time to reside on an arc, or dynamic – unbounded token queues on arcs with no ordering where each token carries a tag to identify its role in the computation. The dataflow graph can be specified graphically or textually and its execution occurs asynchronously on the basis of some possible *firing rules* for actors that characterize the static or the dynamic models. Although several interesting proposals based on dynamic models have been formulated in the past [17]–[19], it preserves, through the conventional processor cycle, the von Neumann model at some lower level of its implementation, making it very difficult to directly implement data flow graphs in hardware. Furthermore, given the presence of the coarse grain computation in the dynamic architectures, the context switching still constitutes a heavy limit in performance [2]. Regarding the static model, as it exposes a program parallelism at a very fine grain, it has received several criticisms too. A drawback of the model is that, since the task switching occurs at the instruction level, it cannot take advantage of the instruction level locality which is present in the programs because of its fine grain computation [20]. The objection is that the overhead of fine grain instruction scheduling prohibits the attainment of acceptable efficiency. Additionally, the very fine grain parallelism of dataflow has proved a disadvantage in the realization of dataflow machines [21]. Interestingly, the same very fine grain parallelism makes dataflow approach attractive when applied to FPGA-based computing machines [8]. Indeed, the static model, in contrast to the greedy scheduling policy embodied in the dynamic model – ”execute whenever data is available”, that is inadequate in many circumstances [2], forms a very natural model of computation for systems with many-core-based chips. Moreover, because its fine grain nature, a static dataflow graph is well suited to be mapped directly onto thousands of simple and identical computing units that cooperate to its execution [9].

B. *Design directions*

There are at least six reasons that have motivated this research. First, the demand to directly map in hardware dataflow graphs in dataflow mode (asynchronous execution); second, the need to dispose of a simple dataflow control and actor firing mechanism at a minimal hardware cost; third, the requirement to reduce the continual LOAD and STORE operations and the coherency wall augmenting, thus, performance [22]; fourth, the expediency to employ a high-level programming language also as assembly language for the machine dataflow processor; fifth, the necessity of a good trade-off between the high-level programming model chosen and the specific hardware mechanisms which implement the processor; sixth, the need to overcome the power wall limit in clocked processors; lastly, the necessity of responding to the more and more increasing request of new data driven/big data applications. To meet these requirements, the adopted methodology for the dataflow machine processor has been the co-design approach between the dataflow execution model and the functional programming model as shown in Figure 1, given their strong relation.

Indeed, the lazy and eager evaluation are two computation methods for executing functional programs. While the computation of programs in the lazy evaluation mode is driven by the need for function argument values – *demand driven*, the execution in the eager evaluation mode is driven by the availability of the function arguments – *data driven*. Moreover, since functional languages are referentially transparent, programs written in these languages can be considered static objects. This means that an expression in a functional language depends on the meaning of its component subexpressions and not on the history of any computation performed prior to the evaluation of that expression. As a result, a dataflow graph can be created in demand driven mode with a functional language, its execution can happen in data driven mode by means of a dataflow machine. Among functional languages, the Bakus [23] FP programming style has been chosen instead of lambda style (like LISP) given the fact that, while the latter focuses on combining objects, the former focuses on combining programs using the rule of metacomposition [24] – new functions are produced applying functionals (i.e. combining forms) to functions, which change small programs into larger ones. In our case, the adopted language in FP programming style has been CHIARA [25]. Regarding dataflow models, the *homogeneous* High Level Dataflow System (*h*HLDS) model has been chosen because it only admits actors with two input and one output arcs – homogeneity of I/O conditions, and data tokens – homogeneity of tokens. This allows actors and link of a dataflow program graph to be directly mapped onto and executed in hardware without needing to any control token.

C. The hHLDS model

High-Level Dataflow System (HLDS) [26] is a formal model that describe the behavior a directed dataflow graphs where nodes are operators (actors) or links (places to hold tokens) that can have heterogeneous actor I/O conditions. Nodes are connected by arcs along which tokens (data and control) may travel.

Since in the classical model [27] actors have heterogeneous I/O conditions and token with different arities due to difference between control and data tokens, heterogeneity conditions constitute the main obstacle in designing reconfigurable machines. The mapping and execution of chunks of dataflow program graphs directly onto hardware in reconfigurable computing requires, in general, the partial or total reconfiguration of the building-block device due to the complex connection management among actors, the control software to manage different arities, and the contest switching time. In contrast, the *homogeneous* HLDS (*h*HLDS) resolves these drawbacks. The *h*HLDS model describes the behavior of a static dataflow graph imposing homogeneous I/O conditions on actors but not links. Actors can only have exactly one output and two input arcs and consume and produce only data tokens, links represent only connections between arcs. Since *h*HLDS actors cannot produce control tokens, merge, switch, and logic-gate actors [28] are not present. While actors are determinate, links[1] may be not determinate. In contrast, these features simplify the

[1] In *h*HDLS there exist two types of links: i) Joint links, which represent a place where two or more output arcs can coexist and ii) Replica links, which are similar to joint links but have only one output arcs. In the case of Joint links the output arc (among the several available), where the token will travel, is unpredictable.

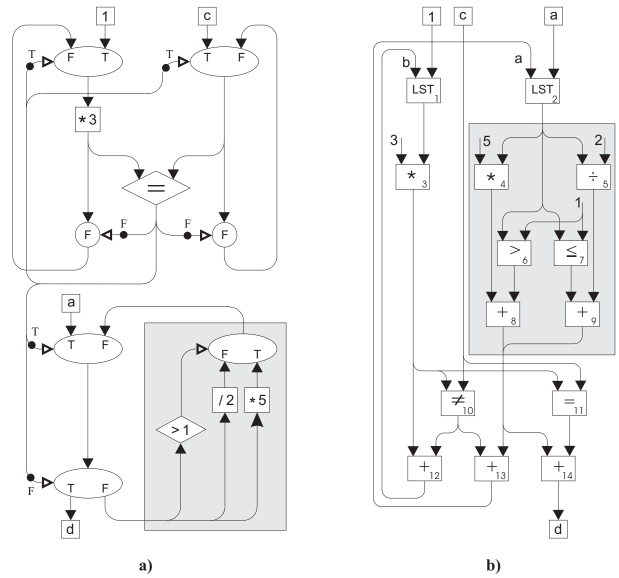


Fig. 2: DPGs for the: **a)** classical model; **b)** *h*HLDS

design of a dataflow execution engine chip using only identical computing units and one type of connection among them. In addition, despite the *h*HLDS model simplicity, it is always possible to draw dataflow program graphs (DPGs) which are determinate and well-behaved where:

- actors fire when their two input tokens are valid, i.e. able to fire an actor, and no matter if their previous output token has not been consumed. In this case, the new token shall replace the previous one. In a system that allows the flow of only data tokens, this property is essential to construct determinate cycles (loops);
- to execute a program correctly, only one way token flow is present as no feedback interpretation is needed;
- no synchronization mechanism needs to control the token flow, thus the model is completely asynchronous.

1) difference between the classical model and the hHLDS: To better understand their difference, let us consider the following program:

```
input (a, c)
   b := 1;
   repeat
      if  a > 1 then  a :=  a \ 2
      else   a :=  a * 5
      b := b * 3;
   until b = c;
   d := a;
output (d)
```

where *a* and *c* are the input variables and *d* is the output variable. Figure 2.a) and Figure 2.b) show respectively the equivalent well-behaved DPG in the classical model and *h*HLDS for the above program. The DPG inside the grey rectangle represents the body of the program, the conditional structure *if ... then ... else*, while the other part represents the iteration control.

Observing the DPG in Figure 2.a, first of all we note that

besides having heterogeneous I/O conditions of actors, it has heterogeneous links and values (data links to hold data tokens and control links to hold control tokens). To comprehend how this DPG works, we have to follow the flow of two different kinds of tokens along graph where actors can have different numbers of input and output arcs, and consume and produce different kinds of tokens. Then, the initial behavior of actors like F-gate, Switch, and Merge depends on their position in the DPG rather than on the program input values. We point out that the initial control tokens for the Switch and Merge actors are automatically present on their control arcs and have different values although they share the same control link. Furthermore, these control values, even though they might be deduced, are not a program input but a programmer's trick to allow the computation to start correctly. Besides, not all the functions associated with actors are defined in the same domain and assume value in the same codomain. For example, if **Z** represents a subset of real numbers, **B** represents the set of boolean values, and **W** represents the set $\mathbf{Z} \times \mathbf{B}$, we can observe that, while the function associated with an arithmetical actor is defined and assumes a value in **Z**, the function associated with the actor Decider is defined in **Z** but assumes a value in **B**, and the functions associated to the actors Merge, Switch, and T-gate are defined in $\mathbf{Z} \times \mathbf{B}$ but assume a value in **Z**. Finally, the implicit definition of static dataflow, for this model, requires that each arc is constituted by a pair of arcs: value and signal [29], to control the condition that an actor cannot be enabled unless all of its output arcs are empty. This control, obtained by means of a data/acknowledgement mechanism between actors, augments the communication overhead and in a certain way complicates the dataflow model because it originates two opposite token flows for a same computation, one for data and another for signals.
In opposite, the DPG shown in Figure 2.b employs only actors with homogeneous I/O conditions; its initial behavior only depends on the program input values rather than on its position in the graph; the well-behavedness check is guaranteed without generating opposite token flows; finally, actors consume and produce only data tokens.

2) Fundamental structures in hHLDS: In the *h*HLDS model actors and links are connected to form a more complex DPG. However, the resulting DPG may be not determinate if cycles occur because no closure property can be guaranteed [30]. This happens for sure when the graph includes joint links, which are not-determinate. In the case when the DPG results to be determinate, we name it macro-Actor (mA). Obviously, an mA is characterized by having I(mA)> 2 and O(mA)≥ 1 where I(mA) is the number of input arcs (in-set) of mA and O(mA) is the number of output arcs (out-set). Here we only report the fundamental ones that allow the creation of more complex structures, i.e., TEST, COND, and IT_R macro-Actors.

The macro-Actor TEST. The simplest relational structure is the mA TEST. It is an example of data-dependent DPG. When coupled to its complement $\overline{\text{TEST}}$, it forms a fundamental building-block to create conditional and iterative mAs. TEST is represented by a determinate and well-behaved mA with

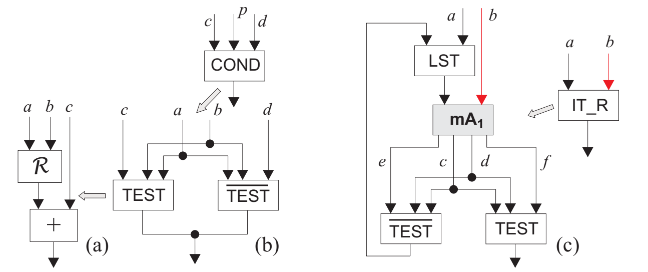


Fig. 3: The basic macro-Actors (mAs) in D#: (a) TEST, (b) COND, (c) IT_R

in-set = 3 and out-set =1 and formed connecting the relational actor $\mathcal{R}$ to the actor that performs the arithmetic operator **+** as shown in Fig 3(a). If $a, b, c \in \mathbb{R}$, its semantics is:

$$\mathsf{TEST}(a,b,c) = \begin{cases} c & \text{if } a\ \mathcal{R}\ b \text{ is satisfied} \\ \perp & \text{otherwise} \end{cases}$$

$\perp$ stays for not valid value. When the actor $\mathcal{R}$ satisfies its relation on the tokens *a* and *b*, it produces a token that has the data-value 0 (zero) and the validity[2] "valid", thus the operation produces the token c. When the relational actor $\mathcal{R}$ does not satisfy its relation, it produces a token that has the data-value don't-care and the validity "not valid".

The macro-Actor COND. The simplest conditional structure is the mA COND, shown in Fig. 3(b). It forms the building-block to create more complex conditional structures. COND is represented by a determinate and well-behaved mA with in-set = 4 and out-set =1. It is formed connecting the two mAs TEST and $\overline{\text{TEST}}$ with a link Joint. If $a, b, c, d \in \mathbb{R}$ and $p = a\,\mathcal{R}\,b$. Its semantics is:

$$\mathsf{COND}(a,b,c,d) = \begin{cases} c & \text{if } a\,\mathcal{R}\,b \text{ is satisfied} \\ d & \text{otherwise} \end{cases}$$

The macro-Actor IT_R. The iterative data-depend structure is the mA IT_R. It constitutes the building- block to create more complex data-dependent iterative structures. It is represented by a determinate and well-behaved macronode with in-set = 2 and out-set =1. IT_R is formed connecting the two mAs TEST and $\overline{\text{TEST}}$, an arithmetic actor or a macro-Actor mA$_1$, and the actor **LST** (loop start) as shown in Fig. 3(c). The LST semantics is: it selects the right token the first time which is fired, the left token otherwise. If $a, b, c, d, e, f \in \mathbb{R}$, its semantics is:

$$\mathsf{IT\_R}(a,b,mA) = \begin{cases} \mathsf{IT\_R} & \text{if } c\ \mathcal{R}\ d \text{ is satisfied} \\ f & \text{otherwise} \end{cases}$$

Observing Fig. 3(c), we point out that, if mA$_1$ (at the center of Fig. 3(c)) is itself an IT_R, the figure represents a determinate and well-behaved nested-data-dependent iterative structure.

D. CHIARA language and the compiling tools

Like in the Backus FP programming style, CHIARA programming system is a tuple (***O***, ***F***, $\mathcal{F}$ **:**, ***D***), where: **O** is a set of objects; **F** is a set of functions (or operators) from objects to objects; $\mathcal{F}$ is a set of functional forms (*functionals*) from

[2]Validity, an intrinsic characteristic of a token, is a binary value whose meaning is: able to fire an actor if valid, unable if not valid

functions to functions; **:** is the application operation; **D** is a set of function definitions. Objects include atoms, sequences and the *undefined* special object $\perp$, called *bottom*, which is used usually to denote errors. Atoms include integer fixed and floating-point numbers, true values, characters and strings. Sequences are denoted with angle brackets. For example,

$$1 \quad <1,2,3> \quad << 1,2 >< 3,4 >>$$

represent three valid objects. CHIARA operators can be elementary or combinators. Elementary operators include commonly used binary, relational operators, and the binary operators loop-start, **LST**, shift-left, **SL**, and shift-right, **SR**. Their peculiarity is that they form the *functionally complete* set – able to generate any other more complex function a program may need by applying the metacomposition rule and are also the actors in the *h*HLDS model. Combinator operators represent functions that affect the structure of the objects on which they are applied. As an example, there are combinators that extract objects out of a sequence, that combine sequences, transpose sequences of sequences, etc. For example, the combinator rotate left *rotl* applied to the sequence `<1,2,3>`, `rotl:<1,2,3>`, returns the sequence `<2,3,1>`. Due to their "transformational" semantics, combinators are not actually executed during the execution of a program, but they are processed during the compilation phase.

Functional forms are used to define new functions out of existing elementary operators and combinators. They include the functionals of FP and some new ones. There is, e.g., *apply-to-all* & that applies a function to all the items in a sequence; *composition* $\circ$ that applies a function to another function; *construction* [ ] that applies a sequence of functions to an object; and so on.

Finally, the *def* construct allows to define a new function using operator, functionals, and other functions already defined. The sample program

```
def max = (GT -> 1; 2)
max:<5, 6>
stop
```

evaluates the maximum between 5 and 6. A detailed description of CHIARA in terms of its objects, functions, functional forms, application operations, and function definitions can be found in [25].

The powerful program algebra of CHIARA is able to extract all parallelism a program has in terms elementary operators, something like

$$(* \circ \&+) :<< 1,2 >,< 3,4 >>$$

As CHIARA programs are variable free, it is possible to easily recognize in the code the functions that only route data to the places where they are consumed and distinguish such code from the one that actually performs computations.

When compiled, a program is first transformed into a DPG, according to the *h*HLDS representation, and then mapped onto the dataflow machine processor, exploiting all its features to get a high performance execution. For example, given the sequence `<<7,3>,<8,4>,<5,6>>`, for each sequence object

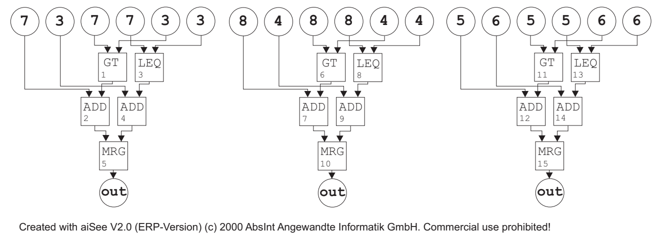


Fig. 4: The DPG for the program $\&max :<< 7,3 >,< 8,4 >,< 5,6 >>$.

we want to determine the maximum value. The corresponding program is:

```
def max = (GT -> 1; 2)
&max:<<7,3>,<8,4>,<5,6>>
stop
```

where the first program line defines the function `max` for a sequence of two elements $< x_1, x_2 >$. Its semantics is: if x_1 is greater than (`GT`) x_2 the combinator form *selector*=1 is applied to $< x_1, x_2 >$, and x_1 is selected; otherwise the combinator form *selector*=2 is applied, and x_2 is selected. The second program line evaluates the maximum for each object of the sequence. `&` is the functional *apply to all* that simultaneously applies the function `max` to all objects of the sequence, making possible the parallel execution of the function `max`. The

TABLE I: The GDPGT for the $\&max$ program

Node#	Func	Left In	Right In	Out
1	GT	%7	%3	2
2	ADD	%7	1	5
3	LEQ	%7	%3	4
4	ADD	%3	3	5
5	MRG	2-4		out
6	GT	%8	%4	8
7	ADD	%8	6	10
8	LEQ	%8	%4	9
9	ADD	%4	8	10
10	MRG	7-9		out
11	GT	%5	%6	12
12	ADD	%5	11	15
13	LEQ	%5	%6	14
14	ADD	%6	13	15
15	MRG	12-14		out

compilation result produces both its graphical representation (Figure 4) and its set of dataflow instructions organized like the row of a table (Table I). The first column contains the number of each actor/computing unit, the second column contains the elementary operator associated to the actor/unit, the third and fourth columns say which is the actor/unit where input token are from – integer numbers mean actor/unit numbers while numbers starting with % mean program input values, the last column says which actor/unit will receive the result – the tag out stays for final value. The nodes MRG (merge) are only virtual but useful because they denote joint points of actor/unit outputs involved in the macro-Actor COND of the *h*HLDS model. Practically, they correspond to physical connections among the computing units that share their outputs.

III. Related work

There exist several researches that investigate new architectural proposals for dataflow processors using FPGA as computation model. However, our dataflow machine is unique with respect to them because its reconfigurable processor executes dataflow program graph contexts only modifying the code of a custom interconnection and the operation codes of the Computing Units, i.e., the actors of the dataflow program graph.

A major recent dataflow project that investigated on how to exploit program parallelism with many-core technology was TERAFLUX. Its challenging goal was to develop a coarse grain dataflow model to drive fine grain multithreaded or alternative/complementary computations employing Teradevice chips [31]. However, the project did not address aspects on how to directly map and execute dataflow program graphs and how to tackle the dark silicon risk and the power/energy efficiency for Teradevices.

Among less recent, but still interesting FPGA-based reconfigurable architectures, we only considered those similar to our dataflow machine. TRIPS architecture [32] is based on a hybrid von Neumann/dataflow architecture that combines an instance of coarse-grained, polymorphous grid processor core, with an adaptive on-chip memory system. TRIPS uses three different execution modes, focusing on instruction-, data- or thread-level parallelism. The WaveScalar architecture [33], on the other hand, totally abandons the program counter. Both TRIPS and WaveScalar take a hybrid static/dynamic approach to scheduling instruction execution by carefully placing instructions in an array of processing elements and then allowing execution to proceed dynamically. However, in our configurable dataflow machine, during the execution of an algorithm, it is not necessary to fetch any instruction or data from memory. The GRD (Genetic Reconfiguration of DSPs) chip [34] is specialized for neural network applications and is constituted by a RISC processor to execute sequential tasks and 15 programmable functional units, DSP processors to execute special tasks, connected in a reconfigurable network of a binary tree shape. In contrast, the dataflow processor can execute both sequential and special tasks and its interconnect is organized like a crossbar. The Morphosys chip [35] is constituted by the 8×8 RC Array, an array of Reconfigurable Cells (SIMD coprocessor) and its Context Memory, a TinyRISC main processor that executes sequential tasks, and a high-bandwidth memory interface. Furthermore, it uses a 2D mesh and a hierarchical bus network. In contrast, our processor exhibits MIMD functionality, its interconnect is like a crossbar, and its context switch is managed by the Kernel Subsystem according to the operations to be executed. For pipeline operations, the context does not change. The FPPA (Field Programmable Processor Array) processor [36] implements a synchronous fixed-point data flow computational model. It employs 16 reconfigurable processing elements (PEs), a programmable interconnect, four 16-bit-wide bidirectional input/output ports, and one 16-bit-wide dedicated output port. The FPPA works in two phases: Configuration, where PEs and programmable interconnects are configured to a specific behavior and to form a processing pipeline; execution, where the program memory specifies sequences of PE and IO module firings individually. In the execution phase, the FPPA reads and processes the input stream of data and writes the result to the programmed output ports. The asynchronous dataflow FPGA architecture [37] describes a low-level application logic using asynchronous dataflow functions that obey a token-based compute model. In this FPGA architecture operators present heterogeneous I/O actor and they operate at cell rather than at computing unit level. Consequently, if the dataflow graph changes, they need a new reconfiguration string. Differently, since all computing units show homogeneous I/O conditions, in our processor, the context switching of a new dataflow program graph only requires the change of the operation and interconnect codes. The WASMII [38] system employs a reconfigurable device to implement a virtual hardware that executes a target dataflow graph. A program is first written in a dataflow language and then translated into a dataflow graph. The partitioning algorithm divides the graph into multiple subgraphs so that deadlock conditions cannot occur. However, the directly mapping of nodes and links that executes a dataflow graph requires the reconfiguration of the device. In contrast, our dataflow machine differs from WASMII because the one-to-one correspondence between actors and computing units and the links and physical connections happens by simply sending the operation codes to the computing units and the configuration code to the custom interconnect.

IV. The Dataflow Machine Architecture

The dataflow machine has a general architecture, shown in Figure 5, that can scale to thousands of nodes. It has been designed to efficiently execute dataflow processes and manage the fine grain computation, typical of the dataflow paradigm. The two major components of the machine are a reconfigurable dataflow computing environment and a communication environment. The reconfigurable environment consists of $2n \times m$ computing units (CUs) which are grouped in *m* processing nodes devoted to execute in hardware DPGs. The communication environment allows data tokens to reach places where they have to be consumed. Moreover, a host supplies all software activities to compile, partition, map, and create the ordered list of DPGs to assign to each processing node, mass storage, and etc.

A. Processing node

It consists of two identical DPG processors and a shared token module that privileges the data locality communication when a DPGs spreads across several nodes, and two graph management modules that store all data and DPG configurations assigned to the node. The three main functionalities that a node actualizes are:

- *Actor Realization*, that creates the one-to-one correspondence between actors of a DPGs and CUs;
- *Token flow Realization*, that allows the I/O token transfer for a DPG execution;
- *Graph Realization*, that turns:

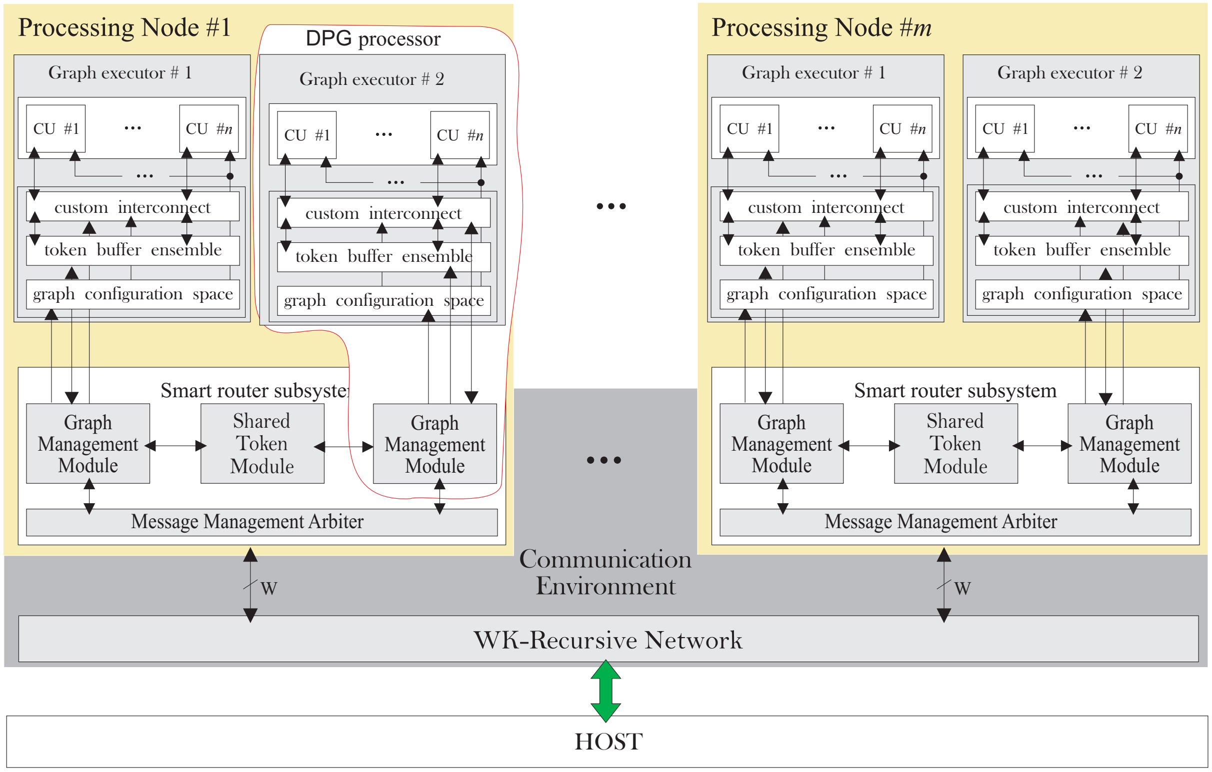


Fig. 5: The general architecture of the dataflow machine

i) DPG arcs into wires that connect CU outputs to inputs of other CUs according to the *h*HLDS;
ii) actor operators into CU operations.

Critical parameters in the dataflow machine design are:

- N_{CU}: the total number of the CUs constituting the reconfigurable environment;
- I_{CU}: the logical and functional intricacy of the CUs;
- In_{tk}: the type of interconnect for the token binary-representation.

n_{CU} depends on the I_{CU} parameter, the number of resources available on the device, and the interconnect complexity $O(n_{CU}^2)$. The ratio between the number of actually available CUs and actors points the way if it is possible to simultaneously map and execute onto the dataflow machine several DPGs of different applications.
When in a DPG the actor number is greater than N_{CU}, first the graph is partitioned in fitting subgraphs, then the subgraphs are mapped onto the DPG processors.

B. Communication environment

Other than the custom interconnect inside a dataflow executor (Executor), other two functional levels belong to the communication environment: i) communication between the two Executors of a node; ii) communication between nodes; these levels constitute the Smart Router Subsystem, and the Internode Network System.

1) Smart Router Subsystem: It consists of the shared token module and the message management arbiter (Arbiter). The shared token module allows the node's executors to quickly exchange tokens when there are data dependencies between the DPGs in execution on them. The Arbiter supervises all communication going outside/in the node. Under its control, entering messages are evaluated, and a decision is taken if to accept or not them. If the node is the message destination, the Arbiter routes the it to the corresponding graph management; otherwise, it forwards the message to the destination according to the its routing policy. The graph management module is the DPG processor control. It is devoted to unpack a message, to supervise the Executor configurations and related I/O data tokens,

2) Internode Network System: Growing the number of processing nodes, the implementation of the interconnection among them requires a hybrid communication solution. In this case, the transfer of tokens messages happens through a WK-Recursive network [39] WK(M, L), where M is the node degree and L is the expansion level. The WK-Recursive network has been chosen for its properties of simple routing policies and highly scalable capabilities. So, the execution model is a hybrid Communicating Dataflow Processes.

C. The DPG processor

The DPG processor is the reconfigurable core of the dataflow machine. It has been tailored to execute in hardware DPGs obtained compiling CHIARA programs. It is composed of an Executor and a graph management module.
The Executor is formed by n identical CUs, a graph setter that implements the graph configuration space, a crossbar interconnect that implement the custom interconnect, and three

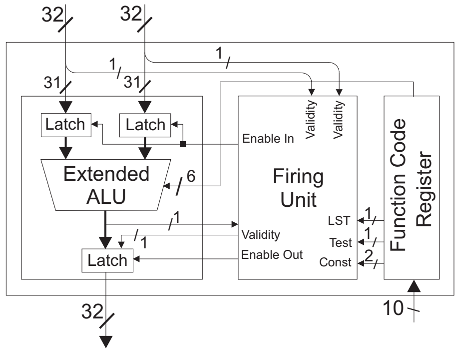


Fig. 6: Computing Unit Architecture.

banks of I/O buffers that implement the token buffer ensemble.

Computing Unit. A CU is formed by an extended ALU – that puts into effect the functionally complete operator set of the CHIARA language, a firing unit – that puts into effect the firing rules of the *h*HLDS model, and two input and one output token latches – to isolate internal CU activities from the others. When a valid token reaches the CU, the firing unit catches its validity. As soon as the partner token arrives, the match occurs, an enabling signal activates the input latches that acquire the two input data tokens so that the ALU operation can take place. After a delay needed to produce the result, the firing unit, according to the executed operation, produces the validity bit for the output token, enables the output latch making available the result token, and resets the values of the two validity bits previously caught. Afterwards a new firing process can start.

Graph Setter. The Graph Setter is a set of two register blocks that stores the DPG ready to be executed. A block is devoted to the interconnect code and the other to the operation code.

Custom Interconnect. The CU interconnect consists of a crossbar that allow to connect any CU output to any other CU input and each register of the Token Ensemble Buffers to the corresponding CU.

Token Ensemble Buffers. This environment is a set of three register blocks with the same memory-size named TOKEN_IN A and B and TOKEN_OUT respectively. The DPG processor can only read the first two registers and write the last one. TOKEN_IN A and B store, for each CU, the right and left token respectively so that token loading operations to/from the processor can properly occur. As token loading can be overlapped to computation operations, the dataflow processor can execute pipeline activities simply receivig/sending data from/to the graph management module.

D. The demonstrator

To verify the correctness and validity of the basic design choices, a demonstrator based on FPGAs has been implemented as well. The demonstrator, whose architecture, shown in Figure 7, is organized like a WK($M = 4, L = 1$). For its implementation, we used a custom board with 5 Altera APEX 20K15-3C devices (Figure 8). 4 of them actualize 4 processing

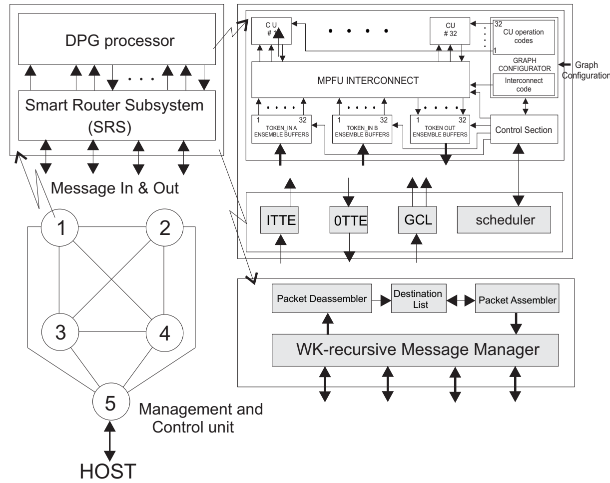


Fig. 7: The architecture of the demonstrator.

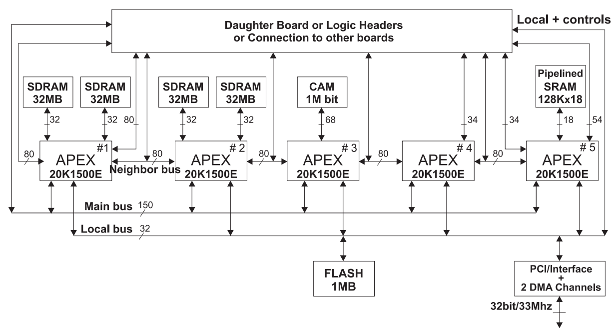


Fig. 8: The demonstrator prototype block diagram

nodes, and the last one actualizes the WK routing facilities and the interface with the host. Intermediate token values which have to flow from a node to another are transferred through the WK router. The executor has 32 CUs which consume two 32-bit tokens (31-bit for data and 1-bit for validity) via a 31-bit fixed-point ALU-multiplier unit, a 10-bit *function code* register, 6-bit for operation code and 4-bit for the firing unit.

V. Performance

To evaluate the D[3]AS co-design approach effectiveness, we have utilized its demonstrator to solve the linear equation system

$$A(n,n)x = B(n) \tag{1}$$

with the Jacobi and Gauss-Seidel iterative algorithms. These two algorithms constitute a good example to approach the same problem in parallel and sequential mode. They calculate an approximation of the exact solution:

$$x_i = \frac{1}{a_{ii}} \sum_{i \neq j} a_{ij} x_j - b_i \quad i = 1...n. \tag{2}$$

To execute the two algorithms, we tailored the corresponding DPG description tables according to the resources of the demonstrator. In this way, we have obtained the configuration files for different values of n.
The execution time for one iteration is given by:

$$T_{iter} = (T_{com} + T_{cal}) \tag{3}$$

TABLE II: Time evaluation expressed in μsec

		Jacobi		Gauss-Seidel	
n	T_s	T_{com}	T_{cal}	T_{com}	T_{cal}
64	83.2	68.38	22.75	22.16	3.20
256	1315.8	547.14	238.23	170.56	65.20
1024	20992.0	7494.25	1436.61	1243.70	612.38

TABLE III: Performance

	Jacobi		Gauss-Seidel	
n	CP	Sp	CP	Sp
64	3.01	0.91	6.92	3.28
256	2.30	1.68	2.62	5.58
1024	5.22	2.35	2.03	11.31

where T_{iter} can be evaluated as a function of n. In fact:

$$T_{com} = t_{CT} + t_{TT} = (n_{CU} * n_{b_{CU}} + n_t * n_{bt}) * t_b * n_s \quad (4)$$

$$T_{cal} = t_{CU} * n_o * n_s \quad (5)$$

where t_{CT} is the configuration transfer time, t_{TT} the token transfer time, t_{CU} the CU execution time, t_b the transfer time for a single byte, n_o the number of sequential elementary operations to upgrade an x_i value, n_s the number of steps to upgrade all the x_i values, n_{CU} the number of processing nodes to be configured, $n_{b_{CU}}$ the number of bytes to configure a node, n_t the number of tokens transferred in a cycle, and n_{bt} the number of bytes per token. t_{CU}, n_{bCU}, t_b, and n_{bt} depend on technology and architecture.

The total execution time in the computational engine is given by:

$$T_{tot} = n_i * T_{iter} \quad (6)$$

where n_i (number of iterations) depends on the initial value set goodness.

The other parameters can be known for each n by the size of the graph description tables and the longest path in the DPG. For instance, it results $n_s = 1$ for the Jacobi method if $n \leq 20$, whereas for the Gauss-Seidel method $n_s = n$, and $n_o = 6 + \lceil \log_2 n \rceil$.

In a sequential environment, the time T_s needed to execute an iteration is given by:

$$T_s = k_1 * n^2 + k_2 * n \quad (7)$$

where k_1 and k_2 depend on the environment and can not be exactly evaluated a priori. However, with a processor power equal to 1.4 Gflops, the minimum values for k_1 and k_2 can be set at 20 nsec.

With $t_{CU} = 15$ nsec, $t_b = 5$ nsec, and $n_{bt} = 4$, Table II shows the values of T_{com} and T_{cal} for several values of n. Due to fine grain dataflow operations and limited resources in terms of CUs per FPGA, most of the time is spent in communication. Some performance indices defined to compare the two methods and evaluate the proposed architecture are shown in Table III. In particular, CP is the communication penalty defined as the ratio between Equation 4 and 5 and speedup Sp is the ratio between the Equation 7 and 3.

VI. Concluding remarks

In this paper we have presented the architecture of a dataflow machine that executes dataflow program graphs directly onto a graph executor. The basic building block of this architecture is the DPG processor, re-interpreted in terms of Many-core technology, constituted by thousands of computing units. The processor is programmed to execute dataflow program graphs, included cycles, according to the static model *h*HLDS and the CHIARA language. In a such way we create the one-to-one correspondence between the actors of the model and CUs of the processor, so that no control tokens are required,. To validate the D^3AS prototype design choices, we have employed its demonstrator, based on a Gidel PROC20KE board with 5 FPGAs, which allows the implementation of the hybrid model Communicating Dataflow Processes. To verify our co-design approach, in terms of processor programmability, management of the very fine grain of operations, and asynchronous execution of DPGs, we have solved some systems of linear equations with Jacobi and Gauss-Seidel iterative algorithms. Results show that this approach is feasible and promising. In the future our effort will be focused on the work on the design of the CU floating-point part able to execute more complex CHIARA operators by means of the CORDIC [40] methodology.

References

[1] T. Sterling and M. MacDonald, “The realties of high performance computing and dataflow’s role in it: Lessons from the NASA HPCC program,” in *Proc. of the IFIP WG10.3 on Architectures and Compilation Techniques for Fine and Medium Grain Parallelism*, C. M., E. K., and G. J.L., Eds. Orlando, FL: North-Holland, Jan. 20-22, 1993, pp. 165–176.

[2] D. Culler, K. Schauser, and T. von Eicken, “Two fundamental limits on dataflow multiprocessing,” in *Proc. of the IFIP WG10.3 on Architectures and Compilation Techniques for Fine and Medium Grain Parallelism*, C. M., E. K., and G. J.L., Eds. Orlando, FL: North-Holland, Jan. 20–22, 1993, pp. 153–164.

[3] M. J. Flynn, O. Mencer, V. Milutinovic, G. Rakocevic, P. Stenstrom, R. Trobec, and M. Valero, “Moving from petaflops to petadata,” *Commun. ACM*, vol. 56, no. 5, pp. 39–42, May 2013.

[4] R. Miller and J. Cocke, “Configurable computers: A new class of general purpose machines,” in *International Symposium on Theoretical Programming*, ser. Lecture Notes in Computer Science, A. Ershov and V. A. Nepomniaschy, Eds. Springer Berlin / Heidelberg, 1974, vol. 5, pp. 285–298.

[5] G. Estrin, “Reconfigurable computer origins: the ucla fixed-plus-variable (f+v) structure computer,” *Annals of the History of Computing, IEEE*, vol. 24, no. 4, pp. 3–9, Oct 2002.

[6] R. Weber, A. Gothandaraman, R. Hinde, and G. Peterson, “Comparing hardware accelerators in scientific applications: A case study,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 1, pp. 58–68, Jan 2011.

[7] G. Theodoridis, D. Soudris, and S. Vassiliadis, “A survey of coarse-grain reconfigurable architectures and cad tools,” in *Fine- and Coarse-Grain Reconfigurable Computing*, S. Vassiliadis and D. Soudris, Eds. Springer Netherlands, 2007, pp. 89–149. [Online]. Available: http://dx.doi.org/10.1007/978-1-4020-6505-7_2

[8] M. Milutinovic, J. Salom, N. Trifunovic, and R. Giorgi, *Guide to DataFlow Supercomputing*. Berlin, DE: Springer, Apr 2015.

[9] L. Verdoscia, R. Vaccaro, and R. Giorgi, “A matrix multiplier case study for an evaluation of a configurable dataflow-machine,” in *ACM CF’15 - LP-EMS*, May 2015, pp. 1–6.

[10] ——, “A clockless computing system based on the static dataflow paradigm,” in *Proc. IEEE Int.l Workshop on Data-Flow Execution Models for Extreme Scale Computing (DFM-2014)*, aug 2014, pp. 1–8.

[11] "Technical challenges of exascale computing," April 2013. [Online]. Available: institute.lanl.gov/resilience/docs/JSR-12-310-Challenges_ of_ exascaleFINAL.pdf

[12] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *IEEE Micro*, vol. 32, no. 3, pp. 122–134, 2012.

[13] J. Howard, S. Dighe, and al., "A 48-Core IA-32 message-passing processor with DVFS in 45nm CMOS," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, feb. 2010, pp. 108–109.

[14] "Task force on next generation high performance computing (hpc): Final report," T. S. of Energy Advisory Board (SEAB), Ed., 2014. [Online]. Available: http://http://energy.gov/seab/downloads/report-task-force-next-generation-high-performance-computing

[15] M. S. et al., "Ubiquitous parallel computing from berkeley, illinois and stanford," *IEEE Micro*, vol. 99, no. PrePrints, 2010.

[16] S. H. Fuller and L. I. Millett, "Computing performance: Game over or next level?" *Computer*, vol. 44, no. 1, pp. 31–38, 2011.

[17] D. Culler and G. Papadopoulos, "The explicit token store," *JPDC*, pp. 289–308, Dec. 1990.

[18] R. Iannucci, "Toward a dataflow/von Neumann hybrid architecture," in *Proc. of the 15th Annual Symposium on Computer Architecture*. Honolulu, Haw.: IEEE Computer Society, May/Jun. 1988, pp. 131–140.

[19] G. Egan, N. Webb, and W. Bohm, "Some architectural features of the CSIRACII data-flow computer," in *Advanced Topics in Data-Flow Computing*, J. Gaudiot and L. Bic, Eds. Prentice Hall, 1991, pp. 143–173.

[20] D. Gajski, D. Padua, D. Kuck, and R. Kuhn, "A second opinion on data-flow machines and languages," *IEEE Computer*, vol. 15, pp. 58–69, Feb. 1982.

[21] P. G. Whiting and R. S. Pascoe, "A history of data-flow languages," *IEEE Annals of the History of Computing*, vol. 16, no. 4, pp. 38–59, Winter 1994.

[22] C. Kozirakis and D. A. Patterson, "A new direction for computer architecture research," *Computer*, vol. 31, no. 11, pp. 24–32, Nov. 1998.

[23] J. Backus, "Can programming be liberated from von Neumann style? a functional style and its algebra of programs," *Communications of the ACM*, vol. 21, pp. 613–641, Aug. 1978.

[24] J. W. Backus, "Is Computer Science based on the Wrong Fundamental Concept of 'Program'? An extended Concept," in *Algorithmic Languages*, J. W. de Bakker and J. C. van Vliet, Eds. North Holland, 1981, pp. 133–165.

[25] L. Verdoscia, M. Danelutto, and R. Esposito, "CODACS prototype: CHIARA language and its compiler," in *Proceedings of the First International Workshop on Embedded Computing.* Tokyo University of Technology, Hachioji, Tokyo, Japan: IEEE Computer Society Press, Mar. 23–26, 2004.

[26] L. Verdoscia and R. Vaccaro, "A high-level dataflow system," *Computing*, vol. 60, no. 4, pp. 285–305, 1998.

[27] J. Dennis, "Data flow computation," in *Proc. of the NATO Advanced Study Institute on Control flow and data flow: concepts of distributed programming*. New York, NY, USA: Springer-Verlag New York, Inc., 1986, pp. 345–398. [Online]. Available: http://portal.acm.org/citation.cfm?id=22086.22093

[28] J. Dennis, J. Fosseen, and J. Linderman, "Data flow schemas," in *International Symposium on Theoretical Programming*, ser. Lecture Notes in Computer Science, A. Ershov and V. A. Nepomniaschy, Eds. Springer Berlin Heidelberg, 1974, vol. 5, pp. 187–216. [Online]. Available: http://dx.doi.org/10.1007/3-540-06720-5_ 15

[29] J. Dennis, G. Gao, and K. Todd, "Modelling the weather with a data flow supercomputer," *IEEE Trans. Computer*, vol. 33, no. 7, pp. 592–603, Jul. 1984.

[30] S. Patil, "Closure properties of interconnections of determinate systems," in *The project MAC Conference on Concurrent Systems and Parallel Computation*. ACM, 1970, pp. 107–116.

[31] R. Giorgi, R. M. Badia, F. Bodin, A. Cohen, P. Evripidou, P. Faraboschi, B. Fechner, G. R. Gao, A. Garbade, R. Gayatri, S. Girbal, D. Goodman, B. Khan, S. Kolia, J. Landwehr, N. M. Lł, F. Li, M. Lujn, A. Mendelson, L. Morin, N. Navarro, T. Patejko, A. Pop, P. Trancoso, T. Ungerer, I. Watson, S. Weis, S. Zuckerman, and M. Valero, "Teraflux: Harnessing dataflow in next generation teradevices," *Microprocessors and Microsystems*, vol. 38, no. 8, Part B, pp. 976 – 990, 2014.

[32] D. Burger, S. W. Keckler, K. S. McKinley, M. Dahlin, L. K. John, C. Lin, C. R. Moore, J. Burrill, R. G. McDonald, W. Yoder, and the TRIPS Team, "Scaling to the end of silicon with edge architectures," *Computer*, vol. 37, no. 7, pp. 44–55, 2004.

[33] S. Swanson, A. Schwerin, M. Mercaldi, A. Petersen, A. Putnam, K. Michelson, M. Oskin, and S. J. Eggers, "The wavescalar architecture," *ACM Trans. Comput. Syst.*, vol. 25, no. 2, pp. 4:1–4:54, May 2007. [Online]. Available: http://doi.acm.org/10.1145/1233307.1233308

[34] M. Murakawa, S. Yoshizawa, I. Kajitani, X. Yao, N. Kajihara, M. Iwata, and T. Higuchi, "The grd chip: genetic reconfiguration of dsps for neural network processing," *Computers, IEEE Transactions on*, vol. 48, no. 6, pp. 628–639, Jun 1999.

[35] H. Singh, M.-H. Lee, G. Lu, N. Bagherzadeh, F. J. Kurdahi, and E. M. C. Filho, "Morphosys: An integrated reconfigurable system for data-parallel and computation-intensive applications," *IEEE Trans. Comput.*, vol. 49, no. 5, pp. 465–481, May 2000. [Online]. Available: http://dx.doi.org/10.1109/12.859540

[36] G. Donohoe, T. Le, D. Buehler, and P.-S. Yeh, "Graphical Design Environment for a Reconfigurable Processor," in *7th Mil/Aerospace Applications of Programmable Logic Devices (MAPLD) International Conference*, Washington, D.C., Sep. 8–10, 2004.

[37] J. Teifel and R. Manohar, "An asynchronous dataflow fpga architecture," *IEEE Trans. Comput.*, vol. 53, no. 11, pp. 1376–1392, 2004.

[38] A. Takayama, Y. Shibata, K. Iwai, and H. Amano, "Dataflow partitioning and scheduling algorithms for wasmii, a virtual hardware," in *Field-Programmable Logic and Applications: The Roadmap to Reconfigurable Computing*, ser. Lecture Notes in Computer Science, R. Hartenstein and H. Grnbacher, Eds. Springer Berlin / Heidelberg, 2000, vol. 1896, pp. 685–694, 10.1007/3-540-44614-1_73. [Online]. Available: http://dx.doi.org/10.1007/3-540-44614-1_ 73

[39] G. Chen and D. Du, "Topological properties, communication, and computing on WK-recursive networks," *Networks*, vol. 24, pp. 303–317, 1994.

[40] R. Goldschmitd, "Applications of division by convergence," Master's thesis, Dept. of Electrical Eng., Massachusetts Inst. of Technology, Cambridge, Mass., June 1964.

Lorenzo Verdoscia Lorenzo Verdoscia received his doctoral degree in Electronic Engineering in 1980 from the University of Naples "Federico II". In 1991 and in 1994 he held a "Visiting Associate in Physics" in the Division of Physics, Mathematics and Astronomy and post-doc position at Caltech and the International Computer Science Institute, Berkeley, California respectively. Currently, he is a senior researcher at the Institute for High Performance Computing and Networking of the National Research Council - Italy. His main research interests include parallel, distributed, and reconfigurable computing, hardware design, dataflow, networks, and functional languages. He has participated in several national and European long-term research projects with universities and industries.

Reviewers:

Australia
Abramov, Vyacheslav; Monash University
Begg, Rezaul; Victoria University
Bem, Derek; University of Western Sydney
Betts, Christopher; Pegacat Computing Pty. Ltd.
Buyya, Rajkumar; The University of Melbourne
Chapman, Judith; Australian University Limited
Chen, Yi-Ping Phoebe; Deakin University
Hammond, Mark; Flinders University
Henman, Paul; University of Queensland
Palmisano, Stephen; University of Wollongong
Ristic, Branko; Science and Technology Organisation
Sajjanhar, Atul; Deakin University
Sidhu, Amandeep; University of Technology, Sydney
Sudweeks, Fay; Murdoch University
Austria
Derntl, Michael; University of Vienna
Hug, Theo; University of Innsbruck
Loidl, Susanne; Johannes Kepler University Linz
Stockinger, Heinz; University of Vienna
Sutter, Matthias; University of Innsbruck
Brazil
Parracho, Annibal; Universidade Federal Fluminense
Traina, Agma; University of Sao Paulo
Traina, Caetano; University of Sao Paulo
Vicari, Rosa; Federal University of Rio Grande
Belgium
Huang, Ping; European Commission
Canada
Fung, Benjamin; Simon Fraser University
Grayson, Paul; York University
Gray, Bette; Alberta Education
Memmi, Daniel; UQAM
Neti, Sangeeta; University of Victoria
Nickull, Duane; Adobe Systems, Inc.
Ollivier-Gooch, Carl; The University of British Columbia
Paulin, Michele; Concordia University
Plaisent, Michel; University of Quebec
Reid, Keith; Ontario Ministry og Agriculture
Shewchenko, Nicholas; Biokinetics and Associates
Steffan, Gregory; University of Toronto
Vandenberghe, Christian; HEC Montreal
Czech Republic
Kala, Zdenek; Brno University of Technology
Korab, Vojtech; Brno University of technology
Lhotska, Lenka; Czech Technical University
Finland
Lahdelma, Risto; University of Turku
Salminen, Pekka; University of Jyvaskyla
France
Cardey, Sylviane; University of Franche-Comte
Klinger, Evelyne; LTCI – ENST, Paris
Roche, Christophe; University of Savoie
Valette, Robert; LAAS - CNRS
Germany
Accorsi, Rafael; University of Freiburg
Glatzer, Wolfgang; Goethe-University
Gradmann, Stefan; Universitat Hamburg
Groll, Andre; University of Siegen
Klamma, Ralf; RWTH Aachen University
Wurtz, Rolf P.; Ruhr-Universitat Bochum
India
Pareek, Deepak; Technology4Development
Scaria, Vinod; Institute of Integrative Biology
Shah, Mugdha; Mansukhlal Svayam
Ireland
Eisenberg, Jacob; University College Dublin
Israel
Feintuch, Uri; Hadassah-Hebrew University
Italy
Badia, Leonardo; IMT Institute for Advanced Studies
Berrittella, Maria; University of Palermo
Carpaneto, Enrico; Politecnico di Torino
Japan
Hattori, Yasunao; Shimane University
Livingston, Paisley; Linghan University
Srinivas, Hari; Global Development Research Center
Obayashi, Shigeru; Institute of Fluid Science, Tohoku University
Netherlands
Mills, Melinda C.; University of Groningen
Pires, Luís Ferreira; University of Twente
New Zealand
Anderson, Tim; Van Der Veer Institute
Portugal
Cardoso, Jorge; University of Madeira
Natividade, Eduardo; Polytechnic Institute of Coimbra
Oliveira, Eugenio; University of Porto
Singapore
Tan, Fock-Lai; Nanyang Technological University
South Korea
Kwon, Wook Hyun; Seoul National University
Spain
Barrera, Juan Pablo Soto; University of Castilla
Gonzalez, Evelio J.; University of La Laguna
Perez, Juan Mendez; Universidad de La Laguna
Royuela, Vicente; Universidad de Barcelona
Vizcaino, Aurora; University of Castilla-La Mancha
Vilarrasa, Clelia Colombo; Open University of Catalonia
Sweden
Johansson, Mats; Royal Institute of Technology
Switzerland
Niinimaki, Marko; Helsinki Institute of Physics
Pletka, Roman; AdNovum Informatik AG
Rizzotti, Sven; University of Basel
Specht, Matthias; University of Zurich
Taiwan
Lin, Hsiung Cheng; Chienkuo Technology University
Shyu, Yuh-Huei; Tamkang University
Sue, Chuan-Ching; National Cheng Kung University
United Kingdom
Ariwa, Ezendu; London Metropolitan University
Biggam, John; Glasgow Caledonian University
Coleman, Shirley; University of Newcastle
Conole, Grainne; University of Southampton
Dorfler, Viktor; Strathclyde University
Engelmann, Dirk; University of London
Eze, Emmanuel; University of Hull
Forrester, John; Stockholm Environment Institute
Jensen, Jens; STFC Rutherford Appleton Laboratory
Kolovos, Dimitrios S.; The University of York
McBurney, Peter; University of Liverpool
Vetta, Atam; Oxford Brookes University
WHYTE, William Stewart; University of Leeds
Xie, Changwen; Wicks and Wilson Limited
USA
Bach, Eric; University of Wisconsin
Bolzendahl, Catherine; University of California
Bussler, Christoph; Cisco Systems, Inc.
Charpentier, Michel; University of New Hampshire
Chong, Stephen; Cornell University
Collison, George; The Concord Consortium
DeWeaver, Eric; University of Wisconsin - Madison
Gans, Eric; University of California
Gill, Sam; San Francisco State University
Hunter, Lynette; University of California Davis
Iceland, John; University of Maryland
Kaplan, Samantha W.; University of Wisconsin
Langou, Julien; The University of Tennessee
Liu, Yuliang; Southern Illinois University Edwardsville
Lok, Benjamin; University of Florida
Minh, Chi Cao; Stanford University
Morrissey, Robert; The University of Chicago
Mui, Lik; Google, Inc
Rizzo, Albert ; University of Southern California
Rosenberg, Jonathan M. ; University of Maryland
Shaffer, Cliff ; Virginia Tech
Sherman, Elaine; Hofstra University
Snyder, David F.; Texas State University
Song, Zhe; University of Iowa
Wei, Chen; Intelligent Automation, Inc.
Yu, Zhiyi; University of California

Authors of papers are responsible for the contents and layout of their papers.

Welcome to IPSI BgD Conferences and Journals!

http://tar.ipsitransactions.org

http://www.ipsitransactions.org

CIP – Katalogizacija u publikaciji
Narodna biblioteka Srbije, Beograd
ISSN 1820 – 4511
The IPSI BGD Transactions
on Advanced Research
COBISS.SR - ID 119128844