

TLCS: A Digital Library with Resources to Teach and Learn Computer Science

Sébastien Combéfis¹ Guillaume de Moffarts¹
Mile Jovanov²

¹Computer Science and IT in Education ASBL, Belgium

²Faculty of Computer Science and Engineering, Ss. Cyril and Methodius
University, North Macedonia

August 6, 2019

[IOI 2019, Baku, Azerbaijan]

Context

- **Computer science education** tends to be everywhere today
 - Concepts taught in some primary and secondary schools
 - Courses and degrees are offered in higher education
 - Contestants challenge themselves with contests
- Computer science concepts are **not easy to learn**
Abstraction, algorithmic thinking, creative thinking, etc.
- **Website and application** to help learning have been developed
Not always easy to find, not well advertised, not documented

Digital library

- A digital library (DL) is... (Borgman, 1999)

“a set of electronic resources and associated technical capabilities for creating, searching and using information.”

- ...that is typically “constructed, collected and organised...
...by (and for) a community of users.”

- Very few digital libraries with resources to learn CS do exist
Only some specialised DL for higher education and research

TLCS project

- The **TLCS project** is a database with a frontend to access it

Developed as an online web application

- **Two main goals** for the platform
 - Allows teachers/learners to quickly find relevant resources
 - Get information about how to use those resources

✓ **Setting up a digital library with resources to teach and learn computer science concepts**

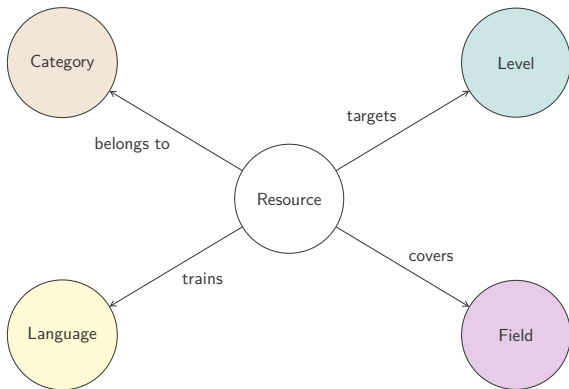
Classifying resources (1)

- Important to **structure the database** to ease searches

Must satisfy the different categories of users: teacher, learner...

- Several possible ways to **classify the resources** are proposed
 - **Category** identifies the kind of service provided
 - **Language** is the trained programming language
 - **Field** is the covered computer science field
 - **Level** is the targeted age group

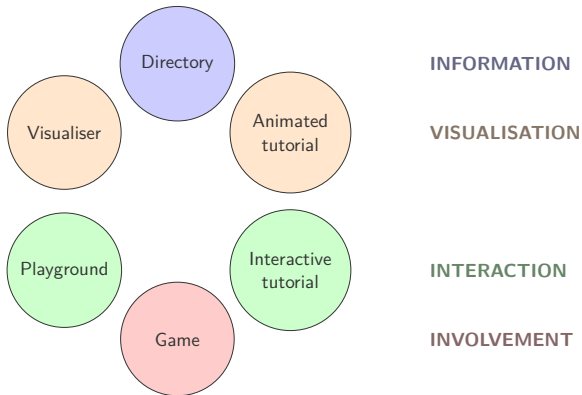
Classifying resources (2)



Category

- **Six main categories** have been identified

Based on the resources analysed in the frame of this work



Directory

- **Directory** allows users to navigate a collection

Resources, technologies, tools, softwares, algorithms...

- Help learners to discover resources related to the **same topic**

Similar in some ways to the “awesome list” movement

- “NoSQL Databases” maintains large list of **NoSQL engines**

Website: <http://www.nosql-database.org>

Directory



Your Ultimate Guide to the
Non-Relational Universe!

[including a historic [Archive](#) 2009-2011]
News Feed covering some changes [here](#) !

NOSQL DEFINITION:Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontally scalable.

The original intention has been modern web-scale databases. The movement began early 2009 and is growing rapidly. Often more characteristics apply such as: schema-free, easy replication support, simple API, eventually consistent / BASE (not ACID), a huge amount of data and more. So the misleading term "*nosql*" (the community now translates it mostly with "not only sql") should be seen as an alias to something like the definition above. [based on 7 sources, 15 constructive feedback emails (thanks!) and 1 disliking comment. Agree / Disagree? [Tell](#) me so! By the way: this is a strong definition and it is out there here since 2009!]

LIST OF NOSQL DATABASES [currently >225]

Core NOSQL Systems: [Mostly originated out of a Web 2.0 need]

Wide Column Store / Column Families

Hadoop / HBase API: Java / any writer, Protocol: any write call, Query Method: MapReduce Java / any exec, Replication: HDFS Replication, Written in: Java, Concurrency: ?, Misc: Links: 3 Books [\[1, 2, 3\]](#), [Guru99 Article](#) >>

MapR, Hortonworks, Cloudera Hadoop Distributions and professional services .

Cassandra massively scalable, partitioned row store, masterless architecture, linear scale performance, no single points of failure, read/write support across multiple data centers & cloud availability zones. API / Query Method: CQL and Thrift, replication: peer-to-peer, written in: Java, Concurrency: tunable consistency, Misc: built-in data compression, MapReduce support, primary/secondary indexes, security features. Links: [Documentation](#), [PlanetC*](#), [Company](#).

Scylla Cassandra-compatible column store, with consistent low latency and more transactions per second. Designed with a thread-per-core model to maximize performance on modern multicore

NoSQL RELATED EVENTS:

- June 26-27 2018 MongoDB World [»](#)

Register your event 4free: [»](#)

NoSQL ARCHIVE



NoSQL FORUMS

- Global NoSQL Forum [»](#)
- Forum Berlin [»](#)
- Forum France [»](#)
- Forum Japan [»](#)

NoSQL NEWS FEEDS

- MyNoSQL by Alex P [»](#)
- On Twitter: nosqlupdate [»](#)
- NoSQL Weekly [»](#) * new *
- HighScalability Blog [»](#)

Visualiser

- Produces static or dynamic **visualisations**

Useful for people sensitive to visual learning modalities

- Help learners to **represent themselves** concepts to learn

Teachers can provide visual examples to learners

- “viSQLizer” illustrates how **SQL SELECT queries** are executed

Website: <http://andmark.no/kristin>

Visualiser

viSQLizer SQL learning tool

SELECT FROM WHERE Erase

SELECT * FROM student

Run new query!

Step 1 of 1

SELECT * FROM student returned the following table:

student

student_no	name	age
99123	Ole Olsen	22
100902	Kristin Annabel	22
150543	Anders Andersen	19

Result table

student_no	name	age
99123	Ole Olsen	22
100902	Kristin Annabel	22
150543	Anders Andersen	19

19

Database tables

course	▼
exam	▼
student	▼
user	▼

★ Saved queries

SELECT * FROM student	▶
SELECT student_no, year, score FROM exam WHERE year > 2013 OR score < 4	▶
SELECT * FROM exam WHERE year BETWEEN 2013 AND 2015	▶
SELECT * FROM student, exam WHERE student.student_no = exam.student_no	▶
SELECT * FROM student JOIN exam ON student.student_no = exam.student_no WHERE exam.year = 2012	▶

Animated tutorial

- **Tutorial** to learn new concepts with visualisations

Can directly present examples with the produced result

- Help learners to directly **see the result** of new concepts

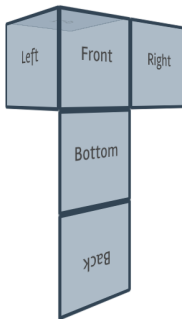
While being guided during the learning thanks to the tutorial

- “Unfolding the Box Model” illustrates **CSS 3D transforms**

Website: <https://rupl.github.io/unfold>

Animated tutorial

Transforms are inherited by children.
You can think of nested elements much like paper folds



Made with ♥ by Chris Ruppel

Playground

- **Execute** and directly get the result of a personal production

Code, problem instance, situation description, model...


- Help learners to experiment with their **own examples**


Without the need to install anything on their computers


- “*RxViz*” execute and show result of **RxJs observable** programs

Website: <https://rxviz.com>

Playground

 **Rx Visualizer**
Animated playground for Rx Observables

Time window (sec) 

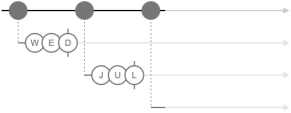


Examples

- Basic interval
- Random error
- Chess game
- Higher order Observable
- Grouped Fibonacci
- Today is...**
- Custom operator
- Mouse move
- Input element
- Pause and resume
- Custom

Feedback

```
1 const { of, interval, range, EMPTY } = Rx;
2 const { delay, take, map, concatMap } = RxOperators;
3
4 const sentence = new Date().toString().toUpperCase();
5 const words = sentence.split(' ');
6 const delayMS = 1000;
7
8 const wordDelay = 1 =>
9   1 === 0
10    ? delayMS
11    : (words[i - 1].length + 1) * delayMS;
12
13 const wordStart = 1 =>
14   1 < words.length
15    ? of(1).pipe(delay(wordDelay(1)))
16    : EMPTY.pipe(delay(wordDelay(1)))
17
18 const wordObservable = word => {
19   const letters = word.split('');
20
21   return interval(delayMS).pipe(
22     take(letters.length),
23     map(i => letters[i])
24   );
25 };
26
27 range(0, words.length + 1).pipe(
28   concatMap(wordStart),
29   map(i => wordObservable(words[i]))
30 )
31
```



Interactive tutorial

- Tutorial with **interactives** related to the concepts
Challenges/problems to be solved by the learners
- Help the learners to directly **practice** with new concepts
Check the understanding of the learners
- “CS Field Guide” online interactive book to **learn CS concepts**
Website: <https://csfieldguide.org.nz>

Interactive tutorial

CSFG Chapters Curriculum Guides Appendices

Search



Algorithms

2.2. Searching

Searching through collections of data is something computers have to do all the time. It happens every time you type in a search on Google, or when you type in a file name to search for on your computer. Computers deal with such huge amounts of data that we need fast algorithms to help us find information quickly.

Lets investigate searching with a game...



Searching Boxes - Part 1

You may have noticed that the numbers on the boxes in the game were in a random order, which meant that finding the target number was basically luck! You might have found it on your first try, or if you were less lucky you might have had to look inside almost all the boxes before you found it. This might not seem like such a bad thing since you had enough lives to look under all the boxes, but imagine if there had been 1,000 boxes, or worse 1,000,000! It would have taken far too long to look through all the boxes and the target number might have never been found.

Now this next game is slightly different. You have fewer lives, which makes things a bit more challenging, but this time the numbers inside the boxes will be in order. The box with the smallest number is on the far left, and the one with the largest number is on the far right. Let's

Algorithms

2.1. What's the big picture?

2.2. Searching

Linear search

Binary search

2.3. Sorting

2.4. What makes an algorithm?

2.5. The whole story!

2.6. Further reading

Game

- Require a big **involvement** of the learners

Challenge to solve given a set of rules and an environment

- Help learners to surpass themselves and to **make progress**

Increased motivation with goals, scoreboards, competitions...

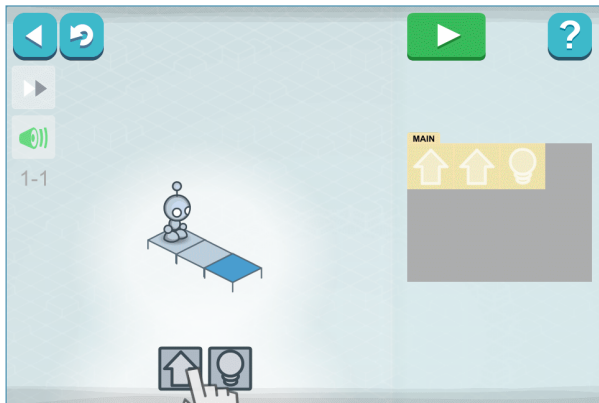
- “Blockly Games” introduces to **basic programming concepts**

Website: <https://blockly-games.appspot.com>

Game

Program Lightbot to **light up** all of the **blue** squares!

Language Select and Full Screen options can be found in the game menu along the right side.



I'm finished with my Hour of Code™

Language, field and level

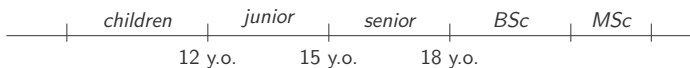
- Three other **classification** ways to help searching resources

Programming language, computer science field and age groups

- Simply **general CS fields** in current version

Database, artificial intelligence, algorithmics, data structure...

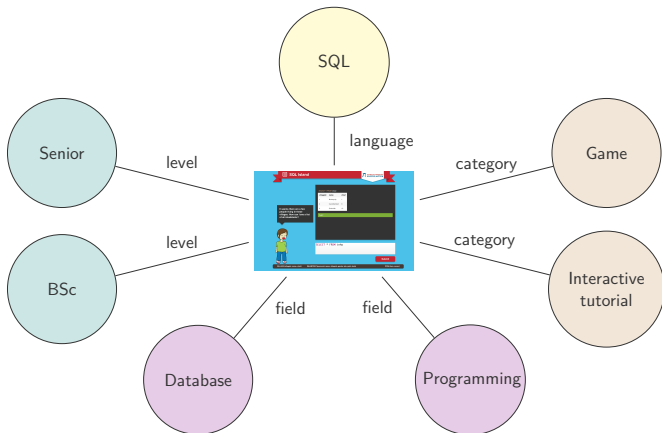
- Most suited **age groups** organised by level of education



Resource example

- **SQL Island** is an adventure game to learn **SQL fundamentals**

Speaking SQL with inhabitants of an island to escape it



Pedagogical information

- Additional informations to help **specific public**

Four main audiences: learner, teacher, researcher and developer

- Optional **pedagogical information** to improve resource use
 - **Prerequisite** mandatory to be able to use the resource
 - **Learning outcomes** list what learner will be able to do
 - **Methodology** explain how the resource can be used
- Two other pieces of **additional information** can be provided
 - **Service** offered by the resource
 - **Reference** to scientific papers

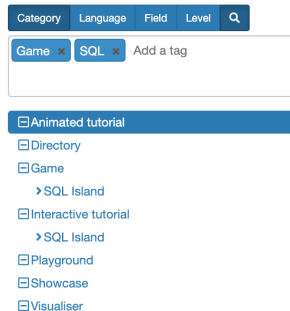
The TLCS platform

- **TLCS platform** used to search for resources

How they can be used to teach and learn CS

- Simple and easy navigation and **search tool**

“Are there any games to learn about the SQL programming language?”



Social and community aspects

- **Content** created and proposed by the community
 - Information must be correct, complete, relevant and up-to-date
 - Review and quality check about entries made by CSITEd ASBL
- Should support knowledge sharing and foster **social interaction**
 - Users will be able to create their own personal tags
 - It will be possible to grade resources with stars
 - More information will be available depending on the user type

Conclusion

- **TLCS** is a digital library with websites and applications

Resources to use to teach and learn computer science concepts

- Proposition of a **multi-criteria** categorisation of resources

Help people to search information relevant to them

- About **twenty resources** have already been encoded

Only in English with the mandatory information

Want to contribute?

Please do! Come and talk to us!

Or just drop me an email: sebastien@combefis.be.