DatVis

System for creating Tabular models for BI in the cloud.

BERGLIND ÓSK EINARSDÓTTIR
BJÖRN INGEMAR ELFSTRÖM
VALDIMAR BALDVINSSON

B.Sc COMPUTER SCIENCE

SPRING 2017

INSTRUCTOR: ÁRNI FANNAR PRÁINSSON
EXAMINER: BIRGIR KALDAL KRISTMANNSSON
DatVis

System for creating Tabular models for BI in the cloud

Berglind Ósk Einarsdóttir
Björn Ingemar Elfström
Valdimar Baldvinsson
Abstract

In this report the B.Sc project DatVis is discussed. This project is in collaboration with the company Viðskipta- og tölvulausnir (VTL), and mainly with the company’s correspondent Hannes Pétursson who is the product owner as well. The project was conducted in the spring of 2017 at Reykjavik’s University by three students of the Computer Science department. DatVis is a system for creating Tabular models from large datasets through a website or by the means of a console application. The Tabular models can then be utilized for Business Intelligence purposes and solutions created with BI tools.
Client Review

VTL has been working in the field of business intelligence for several years. There have been a lot of difficult points in the development, the tools and methods are focused on working with one customer at a time and reusability of code and resources is complicated. We wanted to challenge this and create a platform that makes it easier to manage many customers and at the same time use the power in cloud computing. We were fortunate to get a group of students that were interested in taking on this challenge. We had a clear goal of what problems we wanted to solve but we gave the team freedom for the implementation. We are very impressed on how deeply the group invested in the business domain and the technical implementation exceeded our expectations. We have already tested the application for real customer scenarios and we are very excited to start using the product in production. The quality of the implementation is very professional, the code is clean, the architecture is modular and documentation is accurate, that makes the handover easy and we foresee to start using the system in production in upcoming weeks. The new system opens up for a lot of new opportunities on how to move forward.

Hannes Pétursson
Product Owner - VTL
## Contents

Introduction .......................................................................................................................... 1 
DatVis .................................................................................................................................... 3 
  Data solution ......................................................................................................................... 3 
  Overview .............................................................................................................................. 4 
  Framework ............................................................................................................................ 5 
  Tools ...................................................................................................................................... 6 
  Testing .................................................................................................................................. 7 
  Design ................................................................................................................................... 8 
  Documentation ...................................................................................................................... 11 
Project Management ............................................................................................................. 13 
  Sprints ................................................................................................................................. 13 
  Development ........................................................................................................................ 14 
  Results .................................................................................................................................. 15 
  Hours .................................................................................................................................... 15 
  Story points .......................................................................................................................... 16 
Discussions ............................................................................................................................. 18 
Conclusion .............................................................................................................................. 20 
References .............................................................................................................................. 21
Introduction

This project started with the company Viðskipta- og Tölvulausnir (VTL) which offers general computer services to companies and individuals. It is a small company but in their employees, they have years of experience in areas of computer services, business and software solutions, system maintenance and more. [1]

VTL’s aim is to tailor solutions to their clients and solve problems in the least amount of time and reduce the manual work. From the foundation of VTL there have been several developments and attempts using a wide variety of tools and methods to solve how to work with one customer at a time for a BI solution and keep the option of reusing the code and resources. [2]

Last years the focus at VTL has been on Business Intelligence (BI) which has several approaches. One approach VTL has used was to create BI semantic models that are Multidimensional. With this project, they wanted to reduce the manual work and time that has gone into tailoring their BI solutions, take the work from being on premises to a more cloud environment and with that expand open the possibility to expand their customer base. As well as offering the latest approach in BI solutions, which is Tabular models. Tabular offers a more intuitive option by having a more relational and visual data modeling approach. [3]

The idea and challenge for this type of BI solution was given to the team which consist of three students studying Computer Science at Reykjavik University. Regarding the implementation, the team had complete freedom. From this challenge the team developed the system called DatVis.
The team set a goal after analyzing the needs of the product owner and what the team wanted to accomplish. The goal is and has been from the beginning:

“Design and create a back end system and a website which enables users to transfer large amount of data to BI services hosted in the cloud.”

The vision was to offer a user-friendly way to work with the latest tools and services out there for a better transfer rate and error logging of transfer information.

The final product that the team offers the product owner is this system that creates these Tabular models from a predefined definition from possibly multiple datasets. What the product owner offers his clients is the access to these models, using whatever data providers they would like (Power BI, Excel etc.). With these models, there comes the ability to create reports of financial statuses in relations with all kinds of information that might affect a business such as weather, currency etc.

This report will in the next chapters go through the system structure, project management, discussions regarding the system and project and concluded with a general conclusion of the project.
DatVis

The system DatVis is a fully functional system. It consists of a front end for customers and administrators of DatVis. The main part is the back end to process data into a form that can be used with BI tools. In the next chapters a more detailed description of the system in general will be given.

Data solution

The objective of this project was to create a system that makes tailored data solutions possible in a more standardized platform that could be used for more than one customer. To understand the use of the system it is important to understand the value of the data solution. [2]

“Business intelligence (BI) is a technology-driven process for analyzing data and presenting actionable information to help corporate executives, business managers and other end users make more informed business decisions. BI encompasses a wide variety of tools, applications and methodologies that enable organizations to collect data from internal systems and external sources, prepare it for analysis, develop and run queries against the data, and create reports, dashboards and data visualizations to make the analytical results available to corporate decision makers as well as operational workers.”[4]

This definition as put forward by the Essential Guide of TechTarget is also describing well the common challenge in BI technology that happens when it is used by different software companies and/or advisors, it is often “tailormade” according to needs of each customer. DatVis answers this challenge in a way that it creates a platform that makes it easier to manage many customers and at the same time use the power in cloud computing. [2]
Overview

From the start a diagram was made to represent the transfer process of the data. This overview was created to get a better idea of the steps from data collected to working with a Tabular model in BI. The overview picture displays the process of the project and the connecting components.

The blue box represents possible data sources. Data sources being for example accounting data from dkSystems, weather information from the Icelandic Met Office or currency from The Central Bank of Iceland. The yellow box represents the now created DatVis system that connects to Microsoft SQL Azure, that is represented with a green box. The purple cloud represents Azure Analysis Services and the creation of the Tabular model from DatVis defined template using SQL Server Data Tools (SSDT). At last the red box represents the tools to use with the Tabular model and possibility of defining and molding the data into a more visual outcome in reports form.

Figure 1 Overview of the project’s general process.
Framework
The structure of the system depends largely on services provided by Azure. The approach chosen is largely based around storage containers (Blob) and storage queues hosted in the cloud in order to transfer data over encrypted channels.

The picture below is a representation of the DatVis pipeline, entirely contained on Azure and the process the data goes through.

![DatVis Pipeline Diagram](image)

Figure 2: A overview of the Azure pipeline for DatVis using REST.

It does not go into all the steps of the implementation, just a simplified display of the procedures. These are the steps and their descriptions:

1. – 2. Data source (e.g. Excel) is transferred to storage containers (Blob) through a REST API.
2. When the data source has been transferred to the container the REST sends a message to a Storage Queue.
4. – 5. The Queue consumer pulls messages from the Queue which decides which data source hosted at a container it should process.
6. After the Queue consumer, has processed the data it is inserted into a Azure SQL database.
7. – 8. After the Queue consumer, has successfully inserted the data to the SQL database a tabular definition is created on Azure Analysis Services by the means of a Runbook.

**Tools**

For the development of DatVis many tools were utilized. The team wanted to get the most out of the experience of creating a system from scratch so it was decided to embrace the unknown and choose the latest versions of the tools available. Some tools and services are still in development and have been updated during the project’s lifetime, which created a challenge at times for the team.

For the front end of the website, or the part that users interact with, the three main languages were applied, HTML, CSS and TypeScript. The framework Bootstrap was used as well. For the overall build of the web application Angular 2.0, or Angular, was chosen, which is a single page application framework. It was the choice at the time since it was the latest version. It offers the maximum speed possible on the web platform today and is very popular. Furthermore, we chose to bundle the application using Webpack. [5]

For the back end, or the data access layer, the cross-platform .NET Core 1.1 was chosen. The language C# was used. Since .NET Core is cross platform there is no specific IDE needed, so each developer of the project used an integrated development environment (IDE) they felt comfortable working with. In the end, all team members used Visual Studio Code. [6]
The project came with the opportunity to work with Microsoft Azure. Azure offers a collection of integrated cloud services that the team used to build the application with the help of global networks of datacenters. The product owner supplied each team member with an account with credits to use for setup of servers and whatever was needed. [7]

For authentication the team set up a OpenID Connect and OAuth Framework for ASP.NET Core called IdentityServer4. The framework has access tokens for APIs that are issued for clients. [8]

For the Tabular models the team used Tabular model Scripting Language (TMSL) which was new to all. It is the command and object model definition syntax for Analysis Services tabular model databases. It is a JSON-based statement script based on tabular model semantics (composed on tables, columns and relationships). [9] For the data models the Data Analysis Expressions (DAX) was used to build formulas and expressions in Analysis Services, Power BI and Excel. [10]

Another tool that team had to learn how to use and should be mentioned is the SQL Server Data Tools (SSDT). It is a modern development tool to build SQL Server relational databases or Analysis Services data models. With it you can design and deploy any SQL Server content type. [11]

**Testing**

It was important to the team to get feedback from the actual user of the system. For DatVis user testing there was no official testing environment other than the product owner went through each step of given instruction guides. He gave the team feedback and time was spent on cultivating some of the steps further while he was executing the system setup. It was a great help to have him go through these steps and in the end, he had the necessary services and connections (mostly Azure) setup for the system DatVis.
System error testing was implemented during the development stages of the project. It was systematically done when new features were added. Error messages were in different form from logging, Toaster, Exceptions and email sent. These tests were done by the team.

Regarding unit testing the front end the framework *Jasmine* and the test runner *Karma* was used. For testing framework of the back end .NET Core’s *xUnit* testing was used. Unit tests are in place for the front end and back end to some extent. Of course, code is never fully unit tested and tests should be added as this is a project that will be used and live on more than the initial development phase.

**Design**

Work on the design and structure of the system began in the first weeks. The team wanted the design to be functional and the structure simple. The user stories helped with analyzation of what was needed.

The team chose a name for the system based on it being the combination of the words data and visibility, making data more visual for the customer. The logo came from the fact that the system will be mostly be used for turning accounting data (bookkeeping) to digital information in the cloud.

It was established right away that there should be two user interfaces or dashboards, one for administrators and another for customers. The two dashboards have the same look except the administrator has more access to features, e.g. create companies.

The initial prototypes have evolved greatly from the beginning but the result is more simple than first anticipated. The team tried in the design to make it as easy as possible for the user to use, e.g. there is translation available from English to Icelandic and a good filter for searching through items such as data sources.
In the next four pictures below you can view the displays for front page, dashboard, transfer information and source creation. These displays are to give the idea of how the website and user interface looks like. Not all views are shown.

The front page is to introduce potential clients to the system and it usages for businesses that want to get better view of their financial status in relations to other information. Most of all the front page is to advertise contact information to potential customers for the system.

Figure 3: Front page of the website for DatVis. Pictures on front are from the website Pixabay.com.
The next picture displays the dashboard of the administrator of DatVis. There he can register a new company and its users. View each company’s contact information and their connection string to their Tabular models.

The next picture displays the transfer information dialog box that a user can chose to view and get information such as warnings of rules that have been broken or if the transfer did not go through. This information is sent to the user in an email when the transfer is completed, or if it is registered as failed then the error message.
The last picture displays the dialog box that user gets when he wants to register a new data source. A pre-defined template is given from DatVis system that he can select from, these templates have been tailored to the customer needs and data format. Rules can be applied to specific columns of the data, e.g. an amount must be greater than a zero.

![Source dialog where data to be uploaded is chosen from templates given and rules set for data transfer.](image)

Figure 6: Source dialog where data to be uploaded is chosen from templates given and rules set for data transfer.

**Documentation**

For the DatVis system six instruction guides were made. The system setup is complex because there are many steps to be implemented in a certain way and explanations needed to be given along the way. Almost all of them are for the main user, or to be precise the employee of DatVis who manages the system (the administrator).

One instruction guide was made for the general customer who uses the dkSystem for accounting. This guide entails a step by step instruction on how the customer can collect data from the dkSystem and put it in a Excel format so it can be
uploaded on the DatVis webpage. The product owner asked for these instructions since there seems to be no such step by step guide in place and he would like to hand it to his customers.

The other five instruction guides are both development documentation and a system handbook. They are in a folder with the project in a Markdown format. They explain step by step what the administrator needs to know to setup the connection and services the system needs, the template definition procedure for the Tabular model and as well how the system is executed and managed in general. The names of these instructions are:
Project Management

In the very beginning of the project the team decided to use the agile method SCRUM. It was chosen mainly because it fit the teams need for a certain structure for planning and organization. Almost all the team work and development was done remotely with team members in three different countries. For the last weeks of the project’s lifetime the team members were all in Reykjavík. The product owner provided great facilities at Sjávarklasinn were the team met and worked every day until the official hand over of the project. The product owner has a office there and it gave the team the opportunity to work closely with him.

Sprints
From the very start the team decided on fixed group work sessions. These sessions were scheduled on Tuesday evenings, Thursday evenings and Saturdays. It was also decided to have a meeting with the instructor every Tuesday evening and with the product owner on Fridays. The team had other obligations such as family, courses, work, and wanted to have a structure in place so that the work would be on track and the project would progress steadily over its lifetime. In the last sprints, the work sessions became a daily occurrence.

During the sprints the team used a Product backlog and a Sprint backlog. The team created the product backlog from user stories they defined themselves. These user stories were prioritized into three groups, A, B and C. A being “Must have”, B being “Good to have” and C being “Nice to have”. Along with the product backlog a Product burndown was kept to make it more visual for the team exactly how many story points were completed and how many story points were left for each completed sprint.

When sprint planning the team selected user stories to move over to the sprint backlog to work on during that sprint. In the sprint backlog the team split each user story into as many tasks as they could, often tasks were added later in the
sprint. The team then did what is called a “Planning poker” (individual time set and then discussed) to decide the time for each sprint. This was done because the team members had never worked together before this project so estimating the task time was tricky. A sprint burndown was kept so the team could see if they were prone to over- or underestimate the time each task took and to see if they were following their ideal effort line for task completion.

Work sessions usually started with a short Standup meeting. At these meetings, the work that was done was discussed, whether someone of the team needed help with a task and then tasks for the session and next days were selected. For these work session, the team used Hangouts, Slack and kept the project schedule on Google Drive.

At the end of every sprint, usually on Friday, a short retrospective was done by the team. This helped the team to get perspective for the next sprint planning session, on Saturdays. It is worth mentioning that the team never had any major setbacks or issues to discuss on these retrospective meetings. What was discussed was usually about tasks taking more time than anticipated and updates on new things a team member had learned regarding the tools or services being utilized.

The team decided to split the project’s lifetime into six sprints. The first 1-5 sprints being 3 weeks each and sprint 6 being 2 weeks.

Development
For the development and version control of this project the team used Bitbucket and Git. Bitbucket was chosen since it offers a free private repository and the team wanted to keep the code so it could be easily accessed and safely stored. Feature branching was applied keeping the master clean for delivery ready.

The team wanted to prevent integration problems and aimed at continues integration. The repository has a webhook to a Jenkins Continous Integration
server hosted by Azure, which starts the test-pipeline consisting of a set of stages. In each stage a single script is executed so that the pipeline is easily modified by modifying the stage script or by adding a stage and correlating script.

The commit stage, which resolves all dependencies, builds the project and runs the unit tests. The built binaries are then pushed to Docker-hub using Docker images if all unit tests succeed. The Docker image is then deployed into production like environments for further testing on other stages. That also means that the product is always ready to ship into production, by simply pulling the latest Docker image, this will all be done with a production stage allowing for delivery by the press of a button, providing a Continuous Delivery.

Results

The team achieved its goal of finishing all A priority stories and some of B priority stories. The work was evenly distributed and in the end, equal hours worked for team members. The team members had different strengths and utilized that as much as they could to the benefit of the project. Explaining and learning new aspects of such a system helped the team members learn a great deal more from this experience than they would have done otherwise.

Hours

The team worked together from the middle of January to the middle of May 2017. With the project description came an estimation of 900 hours of work in all needed to finish the product. With the sprints being 6 the team decided to divide it and have around 150 hours of work for the whole team in each sprint, per team member would be 50 hours. The team was aware of disruptions of exams and projects being due in other courses for sprint 4 but with plan to catch that work up in sprint 5. In the end the team went over the estimation by 183 hours in all.
In the table below is the time summary displayed for each team member and the total times for the whole team.

<table>
<thead>
<tr>
<th>Sprint</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berglind</td>
<td>44</td>
<td>41</td>
<td>49</td>
<td>7</td>
<td>85</td>
<td>138</td>
<td>363</td>
</tr>
<tr>
<td>Ingemar</td>
<td>44</td>
<td>36</td>
<td>49</td>
<td>15</td>
<td>85</td>
<td>137</td>
<td>365</td>
</tr>
<tr>
<td>Valdimar</td>
<td>39</td>
<td>36</td>
<td>56</td>
<td>35</td>
<td>71</td>
<td>118</td>
<td>355</td>
</tr>
<tr>
<td>Total</td>
<td>127</td>
<td>113</td>
<td>154</td>
<td>56</td>
<td>241</td>
<td>393</td>
<td>1083</td>
</tr>
</tbody>
</table>

**Story points**

Story points were used to give user stories a tangible value on how time consuming and difficult they were. The number of user stories grew during the project’s lifetime and in the end, there were 42 stories with an estimation of 245 story points (A, B and C). Of those stories 27 were A priority stories with 149 story points. The team completed all A priority stories as well as 2 B priority stories with 16 story points. That makes the finished product with 165 implemented story points, or 29 stories completed.

The team was unfortunately never on track with the ideal burndown of the product this was probably due to the underestimation of the time needed for the
tasks and mostly because of a learning curve of the new tools and services in place. In sprint 4 the team comes to a halt due to exams and projects hand ins in other courses. This stop was planned and the burndown of the story points went really well in sprint 5, which is worth to mention was the best sprint, estimation wise that the team participated in.

Figure 7: Product burndown for DatVis. Starting at 165 story points. All 27 A priority stories and 2 B stories.
Discussions

The team is very proud and satisfied to be able to handover a fully functioning system ready for deployment for the product owner. The set goal was achieved and the team work was excellent.

The team had a last overall project retrospective meeting were the main topic was what could have gone better. The team tried hard to find issues to discuss and although it may seem like an exaggeration the team did not find anything that they would have done differently as a team knowing what they know now. Of course, one can always do better both as an individual and as a team but we are pleased with the collaboration, the work we have done and the product. The product will always be in a never-ending development stage, which is how it should be. There will always be features to be implemented or a direction for it to take that we cannot see now.

One thing the team did discuss was the unexpected request from the product owner for a console option for uploading data sources. The team is not sure that this would have changed the implementation of the project but if the project had not been a school project and made solely for the product owner the team might have conceptualized the project differently with a more focus on a console aspect.

When it came to learning to work with new tools it was a learning curve and sometimes a challenge, especially with Microsoft Azure. Mostly because development of the Portal is still going on so its use is not as documented as the team would have liked, meaning that no explanations about required information for creation of services is in place. The portal's appearance did change during the project and the team had some trouble getting assistance when not all team members had trouble accessing the Portal.
Since the team lived in three different countries during most of the work it was down to long distance communications and scheduled group work to keep the team going. The team can thankfully say that there was never a problem with working together. Of course, it is always more enjoyable to work in the same space and get to know each other better that way, but the team feels it was not a hindrance to the project having it be a long distance one. With the time difference, other obligations, exams and projects it had been discussed that it might become a challenge but it never seemed to be that way. All team members respected each other’s time and shared the responsibility of the project by working together on it equally.

A good challenge the team faced in the end was having the client working close to us and being very interested in the system. This of course is great and we welcome it and his input helped the team greatly, but with this type of challenge tasks are added and development tends to go further than was planned due to excitement of the possibilities. This collaboration however enabled the team to get a better and more clear idea of what was needed from the system DatVis.

In the end the team is happy with the system and the product owner seems to be very happy with the product and excited to deploy it.

Future progress can be made with the system such as more tabular model features, improving performance or making the solution design phase easier, even automated. There are possibilities to speed up the transfer rate of data more than it is. The team is curious and probably the most excited about the possibility of machine learning (data mining in Analysis Services) and with that have predictive analysis reporting. Regarding making the system more valuable and desirable for a general customer team would like to see the reports the customer can create and manipulate in the DatVis system itself, even have it in a form that is specifically for mobile phone setup.
Conclusion

This report introduces the system DatVis, the team work behind its creation and the overall project management.

In the beginning, we set a project goal and the team is very proud and happy to be able to say that the goal was achieved. The final product is a system that can be used now by the product owner and for the purpose it was created for. Of course, future development and maintenance is required as with all systems. The team is excited to see how DatVis will evolve and what is in store for it and its users.

This project was a challenge in a way that the environment and the tools the team chose to work with was new and sometimes still in a development stage. That created a learning curve and adjustments along the way. The team embraced this since it gave a chance to acquire a more versatile knowledge and experience regarding these different aspects of this final project. The team agrees that it could not have gotten a more invaluable and overall experience with the opportunity to work with front- and back end development, external resources and business intelligence.

The team would like to thank VTL for opportunity to work on this project. Special thanks go to Hannes Pétursson, the product owner, for the collaboration, providing top facilities, delicious lunches and mainly for creating an environment of great enthusiasm and interest for the project.

The team would also like to thank the instructor Árni Fannar Bráinsson, the examiner Birgir Kaldal Kristmannsson and the director of undergraduates Hallgrímur Arnalds for their advice and assistance. The people at Sjávarklasinn for being so welcoming and accommodating.

Last but not least we want to thank our families for being there for us through our studies and this process of finishing our final project.
References


Berglind Ósk Einarsdóttir
Kt. 181284-2699

Björn Ingemar Elfström
Kt. 120290-2749

Valdimar Baldvinsson
Kt. 101081-3469