Paper 005-31

Making your own Balanced Scorecard using the SAS® Information Delivery Portal

Paul Smeekens, OZ zorgverzekeringen
Erwin van Dongen, OZ zorgverzekeringen

ABSTRACT

In 2004, OZ zorgverzekeringen migrated its existing Web applications and SAS® Information Delivery Portal to the new architecture of SAS® version 9 using a central SAS® Metadata Server. With this move, all information flows were centralized in the SAS® portal. Still, the amount of information made it difficult to get a quick look into the performance of the organization as a whole.

In 2005, OZ zorgverzekeringen combined several pieces of information into a balanced scorecard. The Balanced Scorecard has been made by using just AppDev Studio® and SAS/Base®. The scorecard is published within the existing SAS®-portal. Now, on the level of the entire organization as well as per department, figures are being published on the financial, internal business processes, customer, and learning and growth perspectives of the company. The balanced scorecard is giving management at OZ improved insight into how the organization is doing.

INTRODUCTION

OZ zorgverzekeringen is a medium sized health insurance company operating mainly in the south-western part of the Netherlands. With approximately 600 employees, about 600.000 insured persons are being served. Most of the 600 employees work in the main office in Breda but there are also some smaller offices in the towns surrounding Breda. Those smaller offices mainly operate as service and selling points.

OZ zorgverzekeringen had a lot of SAS®/Intrnet-application, static HTML-reports published on the intranet and both reports and applications in the SAS® Information Delivery Portal which they migrated in 2004 to the new architecture of SAS® version 9 using a central SAS® Metadata Server. With this move all information flows were centralized within the SAS® portal. Still, the amount of information made it hard to have a quick glance into the performance of the organization as a whole. In the summer of 2005 OZ zorgverzekeringen started working on a Balanced Scorecard in which several pieces of information is combined.

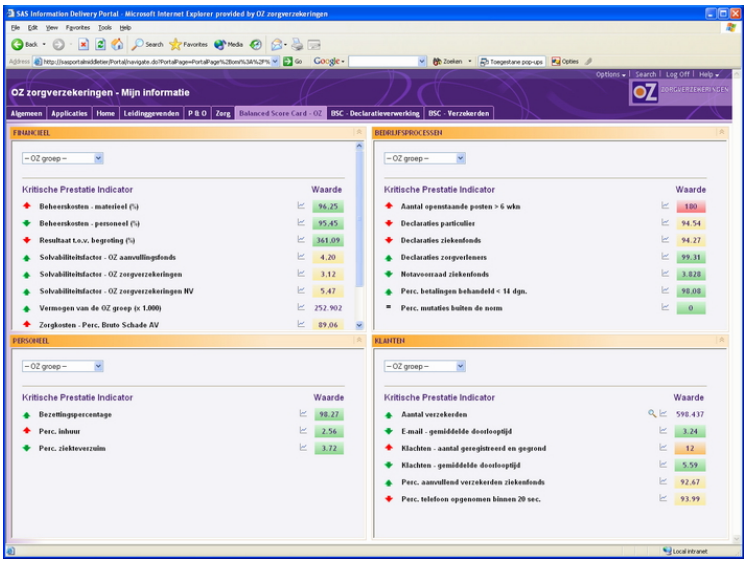

Fig. 1 Overall Balanced Scorecard - OZ

The Balanced Scorecard has been made by using just AppDev Studio® and SAS/Base®. The scorecard is published within the existing SAS®-portal.

With the Balanced Scorecard OZ management now has improved insight in how the organization is doing. Both on the level of the entire organization as well as per department, figures are being published on the:

- financial perspective
- internal business processes perspective
- customer perspective
- learning and growth perspective

These four perspectives make up the framework on which the Balanced Scorecard is based.

In the next few paragraphs both the concepts and the technical issues involved in developing the Balanced Scorecard are being covered.

THE CONCEPT OF THE BALANCED SCORECARD

In the early 90's the Balanced Scorecard[1] surfaced as a response to the growing inadequacy of traditional financial measurement systems to help steer an organization towards its desired future. The Balanced Scorecard complements these financial measures of past performance with measures of the drivers of future performance. The Balanced Scorecard fills the need for making strategy actionable at all levels within an organization as well as the need to bring together all the disparate analytic measurement systems in an organization under one conceptual umbrella. The true benefits of the Balanced Scorecard occur when it is transformed from a measurement system to a management system that helps to clarify, gain consensus on and communicate strategy throughout the company[2].

The framework that makes up the Balanced Scorecard consists of four perspectives on organizational performance. These perspectives are: financial, internal business processes, customer and learning & growth. Using these perspectives a company's strategy is translated into well balanced measures, thus helping people within an organization with the continuous balancing act that is called "doing business" while keeping in mind the organizations' long term goals.

THE DEFINITION PROCESS

In its' day-to-day operations the knowledge centre (KC), a central department responsible for business intelligence within OZ, was presented with a growing number of reports, documents and applications. Some clearly linked to the organizations' long term strategy, others more implicit. Although these information flows were centralized within the SAS® portal it was still difficult to quickly obtain insight in the overall performance of OZ and its' departments. From this came the idea to create a well balanced extract from these sources in the form of a Balanced Scorecard.

This idea was then presented and further developed together with a member of the board of directors thus gaining support at the highest possible level. Since this was also the level at which the business need for such a solution was most apparent, the CFO eagerly became the sponsor of our scorecard. Dealing with the organizations' long term goals he was also the main spokesman in determining the content of the Scorecard at overall organizational level.

The resulting overall page was then fine-tuned during discussions with managers in the organization responsible for the specific KPIs (Key Performance Indicators) that were selected for presentation. They were also the ones capable of defining the ranges for these KPIs. These ranges were then validated against the organizations strategy by our CFO. This way discussion was kept to a minimum while obtaining momentum for the Balanced Scorecard principle was being maximized.

After obtaining visibility for the project by making the overall page available to everyone within the organization using the SAS® Information Delivery Portal, the same steps were taken for a few different organizational units. For this the role of the CFO was taken on by the responsible manager and the fine-tuning was done during discussions with lower level management and employees responsible for the lower level KPIs.

After completing different levels of the Balanced Scorecard it was immediately distributed to every employee within that organizational unit as a separate page within the portal. In addition to this, that same information was made available to the rest of the organization by providing an extra drill-down level in the overall Balanced Scorecard. This technique is explained further in the following paragraph.

THE TECHNICAL PROCESS

In parallel to all this the technique for the front end of the Balanced Scorecard was developed using AppDev Studio® in combination with SAS/Base®. This was done using prototyping starting from a mock-up Balanced Scorecard made with simple HTML working towards a refined solution based upon Portlets and SAS® Stored Processes. Continuously obtaining feedback on the requirements for the solution from the CFO and the managers involved in the project and integrating these into the tool, helped creating the broadly accepted solution.

The automated back end for the Balanced Scorecard was developed using SAS/Base®. By simply adding pieces of code to the regular ETL-processes (extraction, transformation and loading) for the datawarehouse, consistency was guaranteed between the KPIs presented in the Balanced Scorecard and the data presented in applications and reports already in use. The added code did nothing more than calculating the KPI, transforming it into the desired format and uploading it into the SQL datamart. An example of such a piece of source code can be found in appendix 1.

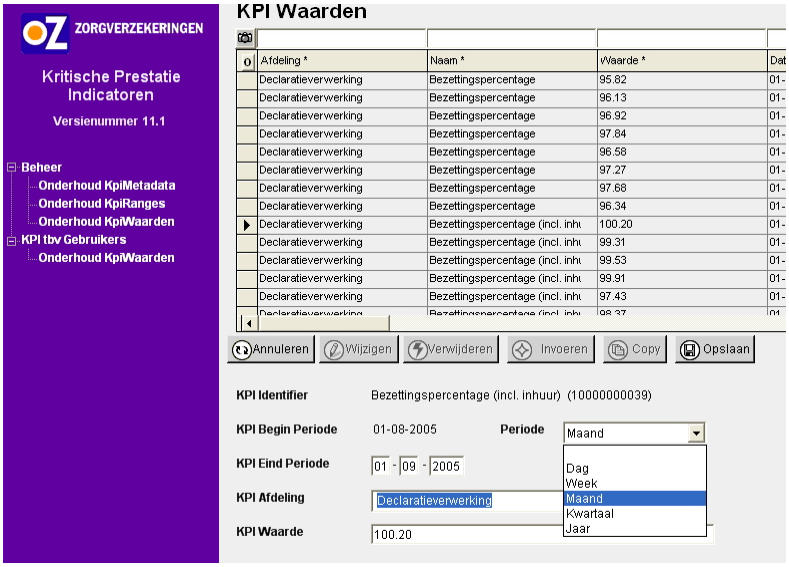

Fig. 2 .NET input application

Since not all measurements can be obtained from automated processes, there was also the need for an application for manually registering KPI values. Because of OZs' experience with using .NET for this kind of input applications, it was decided that this would be the preferred way to go. .NET provides a development environment in which validations needed for this input application can be defined easily. This .NET application provides an interface to the SQL-tables that make up the datamart for the Balanced Scorecard. Detailed descriptions of these tables can be found in appendix 2.

In this .NET application KPIs can be defined and modified, ranges and subranges can be quantified and values for specific KPI's can be provided by authorized users.

OZS' BALANCED SCORECARD

On a daily basis all data is extracted from the SQL-tables making up the KPI datamart. Using SAS/Base® this data is combined and transformed into a single SAS dataset containing all the KPI information combined with metadata, ranges, definitions, etc. This single SAS dataset forms the infomart used by the interface of the Balanced Scorecard.

The interface for the Balanced Scorecard consists of a combination of a portlet together with 3 different Stored Processes. These Stored Processes are registered in the SAS Management Console in order to be available for use in the Information Delivery Portal. The portlet is responsible for determining the validated user as well as taking parameters to be passed on to the Stored Processes. These Stored Processes are:

1. Kpi_main.sas – presenting the KPI's for a department dealing with a specific perspective.
2. Kpi_graph.sas – presenting a graph for the selected KPI.
3. Kpi_definition.sas – presenting the KPIs definition, range, start- and endDate, etc.

The main stored process takes the subject, or perspective, and the department as parameters. From these, using simple SAS/Base® a selection is made from the KPI infomart. The results of this selection are used to dynamically build HTML that is streamed back to the portlet, thus presenting the kpi's.

Fig. 3 The result of kpi_main.sas

Within the generated HTML URL's containing parameters are embedded pointing to Kpi_graph and Kpi_definition as well as a form pointing to Kpi_main.

The embedded form in combination with the combobox containing the departments of OZ provides a way to drill-down or go back to the different levels at which the KPIs are available.

The embedded URL pointing towards Kpi_graph looks like this:

<a href="/SASStoredProcess/do?&_PROGRAM=SBIP://Foundation/OZ/Kpi/Kpi_graph(StoredProcess)
&Kpi_ID=1000000012&paramDept=1DIR00&endDate=31dec2005">
<img src="http://SASPortalMiddleTier/Portal/images/LineChart.gif" border="0">

Where the first line points to the registered Stored Process, the second line contains the parameters to be used, determining which KPI to present for which department and which endDate to use. The last line simply contains an icon for a graph. Off course, this URL is only available within the context of the SAS® Information Delivery Portal.

Again HTML is created in combination with an activeX image. This image is created using ODS in combination with SAS/Graph and device=actXimg. This results in the following graphs.

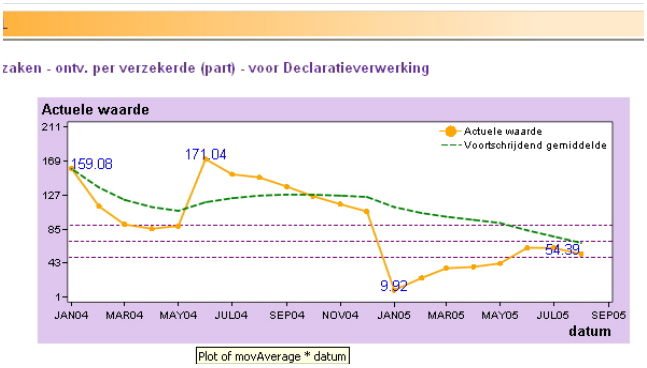

Fig 4. The graphical presentation of a kpi

In this graph the development of the specific KPI is presented along with a moving average, the applicable range values as well as the first, last, minimum and maximum value of the KPI.
The HTML generated by Kpi_main.sas also contains an URL pointing to Kpi_definition.sas. This URL looks like:

<a href="/SASStoredProcess/do?&_PROGRAM=SBIP://Foundation/OZ/Kpi/Kpi_definition(StoredProcess)
&Kpi_ID=1000000012>
Percentage sick leave
****

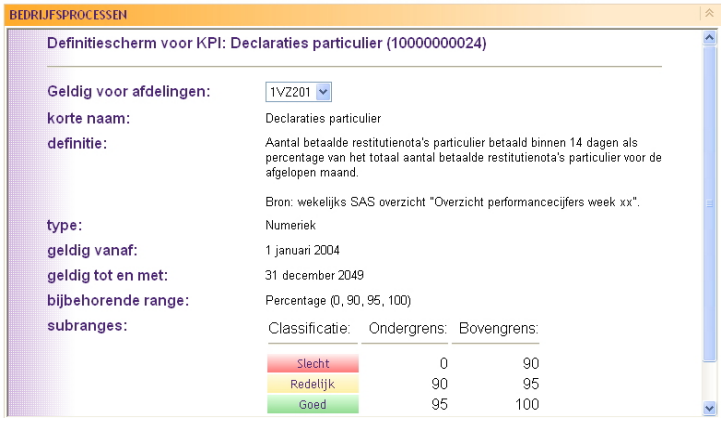

Fig 5. The definition screen

Again the first line points to the registered Stored Process. The second line contains the parameter identifying the KPI. The third line contains the descriptive name of the KPI that is presented in the main screen together with the actual value and color coding. Following this URL the user is presented with a definition of the KPI, the applicable ranges and subranges. etc.

This screen makes sure that everyone viewing the KPI value understands what it means as well as how it was determined. The user can also see what source was used for this measure and with which frequency it is updated.

PITFALLS

First of all it is very easy to get caught up in the hype. The “Balanced Scorecard” has been a buzz-word for several years now. This however should never be the reason for your organization to pursue such a tool. There has to be an actual business need. For us it was the growing amount of information and the need to quickly obtain insight in the overall performance and steering the organization in the right direction.

Another pitfall is to get lost in the theory. All too easily an organization can lose itself in a purely academic exercise, thus losing grip on the business need that is to be fulfilled. The result of this can be endless discussions on definitions of KPIs, the relevance of certain measurements, the influence one has or doesn’t have on certain KPIs, etc.. In our experience it is far better to decide and stick with the decision.

Don't expect a quick fix. Balanced Scorecarding is about obtaining insight in your organization and in how it is working towards its strategic goals. It isn't about the tool, it is about the intelligence and insight your organization obtains from using it. Such a process takes time and effort. For a medium-sized organization like ours this meant starting small, gaining momentum and creating support.

CONCLUSION

The Balanced Scorecard can provide your organization with means to leverage its disparate information flows by combining them using the frameworks' four cornerstones.

By building upon existing reports, applications and other information flows momentum can be created without falling into the trap of endless theoretical discussions. This can only be done in such a manner if the existing information flows have indeed been defined with the organizations' long term strategy in mind.

Once the Balanced Scorecard is in place several steps need to be taken to transform it from a measurement system into a management system that helps to clarify and gain consensus on the company's strategy and communicate it throughout the company[2].

The resulting Balanced Scorecard helps management at OZ to quickly gain insight in overall performance and helps them to focus on problem areas that stand in the way of the long term strategic goals. Using the Balanced Scorecard helps them to determine where to look in detail in the vast amounts of managerial information flows currently available within the organization.

REFERENCES

[1] "Translating strategy into action – THE BALANCED SCORECARD", Robert S. Kaplan and David P. Norton, 1996 – Harvard Business School Press.

[2] "ESSENTIALS of Balanced Scorecard", Mohan Nair, 2004 – John Wiley & Sons.

CONTACT INFORMATION

(Your comments and questions are valued and welcomed

Erwin van Dongen
OZ zorgverzekeringen
Chasséveld 17
4811 DH Breda
The Netherlands
Phone: +31 (76) 5243 325
Email: erwin.vandongen@oz.nl
Web: www.oz.nl

Paul Smeekens
OZ zorgverzekeringen
Chasséveld 17
4811 DH Breda
The Netherlands
Phone: +31 (76) 5243 959
Email: paul.smeekens@oz.nl

APPENDIX 1 AN EXAMPLE OF THE SOURCE CODE USED FOR AUTOMATED LOADING OF A KPI.

In our case the data mart is developed in SQL in order to be able to easily develop a .NET solution for providing manual input. These tables are available to SAS® using SAS/ACCESS to ODBC. This however can just as easily be developed just using SAS datasets.

```
/*------ Modification needed for automated loading of the KPI ----------*/;
/***
  Make a libname to the sql-database used for the kpi's
***/
libname kpi_sql odbc DSN="SQL_KPI" CONNECTION=GLOBALREAD BCP=YES;
libname ebsdata '\\SASPortalBackEndTier\e$\portal\Sasdata\emailbewaking';

/***
  Read the e-mail data and assign it to the proper department
***/
data email_all
     email_deptA;
  length department medewerker $64;
  set ebsdata.embs_mber;

  if _N_ = 1 then do;
    retain patternID;
    patternID = prxparse("/declaratie|verhaal/i");
  end;

  StartDate = put(intnx('MONTH',delivereddate,0,'b'),date9.)||":00:00:00.000";

  KPI05_DepartmentCode = "1DIR00";
  output email_all;

  if prxmatch(patternID, lowcase(department)) gt 0 then do;
    KPI05_DepartmentCode = "1DEC00";
    output email_deptA;
  end;
run;

/***
  Determine the mean time at overall level
***/
proc summary data=email_all missing nway;
  class KPI05_DepartmentCode StartDate;
  var cat06;
  output out=email_all_mean (drop=_type_ _freq_) mean=;
run;

/***
  Determine the mean time at the level of the department “deptA”
***/
proc summary data=email_deptA missing nway;
  class KPI05_DepartmentCode StartDate;
  var cat06;
  output out=email_decl_mean (drop=_type_ _freq_) mean=;
run;
```

```
/***
  Transform the data into the format needed for loading the SQL tables.
***/
data tblKPI05_KpiValues (keep = KPI05_: );
    attrib KPI05_KpiID            length = $11 format=$11.
           KPI05_DepartmentCode   length = $10 format=$10.
           KPI05_StartDate        length = 8   format=datetime22.3
           KPI05_EndDate          length = 8   format=datetime22.3
           KPI05_Waarde           length = $50 format=$50.
           ;
    set email_all_mean
       email_decl_mean;

    KPI05_KpiID = "10000000029";
    KPI05_StartDate = input(StartDate,datetime22.3);
    KPI05_EndDate = intnx("DTMONTH",KPI05_StartDate,1);
    KPI05_Waarde = strip(put((cat06/3600),20.2));
run;

/***
  Delete rows earlier inserted and replace these with the updated values.
***/
proc sql;
  delete from Kpi_sql.tblKPI05_KpiValues
    where KPI05_KpiID in
          (select distinct KPI05_KpiID
           from tblKPI05_KpiValues)
          and
          KPI05_StartDate in
          (select distinct KPI05_StartDate
           from tblKPI05_KpiValues)
           ;
  insert into Kpi_sql.tblKPI05_KpiValues
             (KPI05_KpiID,
              KPI05_DepartmentCode,
              KPI05_StartDate,
              KPI05_DatumEindPeriode,
              KPI05_Waarde,
              KPI05_DatumToevRecord,
              KPI05_UseridToevRecord,
              KPI05_DatumLaatsteMut,
              KPI05_UseridLaatsteMut,
              KPI05_IndVerwijderd
             )
        select
              KPI05_KpiID,
              KPI05_DepartmentCode,
              KPI05_StartDate,
              KPI05_DatumEindPeriode,
              KPI05_Waarde,
              KPI05_DatumToevRecord,
              KPI05_UseridToevRecord,
              KPI05_DatumLaatsteMut,
              KPI05_UseridLaatsteMut,
              KPI05_IndVerwijderd
```

```
        from tblKPI05_KpiValues
        ;
quit;

/***
  Clear the libnames used.
***/
libname kpi_sql clear;
libname ebsdata clear;

/*-------- End of the modification needed for the KPI ----------------------*/
```

APPENDIX 2 THE DATA MODEL USED FOR THE KPI DATA MART.

The data model used for the KPI data mart provides for maximum flexibility thus ensuring a robust design capable of keeping up with future improvements to the balanced scorecard application.

By including sub ranges in a separate table we are able to modify ranges for a specific KPI through time, while still being able to see ranges that were applicable in the past. This way we can continuously redefine the ranges for specific KPIs to keep up with improvement and to steer towards even further improvement for the future. Within this data model ranges can be re-used for several KPIs at once.

The following table descriptions are a simplified version of the actual SQL-table definitions. In reality several database related attributes are part of these tables to facilitate auditing. These attributes include modification_user, modification_datetime, insertion_user, insertion_datetime and is_deleted.

KPI_metadata: table containing the metadata definitions of the KPIs.

Attribute	Type	Length	Label	Content	Range
Kpi_id	char	11	KPI identifier	Automatically generated, unique, internal identifier	000.000.000 – 999.999.999
Kpi_startdate	datetime		StartDate	Datetime indicating the start of the period during which this KPI is valid.	datetime
Kpi_enddate	datetime		EndDate	Datetime indicating the end of the period during which this KPI is valid.	datetime (default 31dec2049:00:00:00.00)
Kpi_name	char	50	KPI name	Descriptive name	
Kpi_definition	char	4096	KPI definition	Descriptive text contianing the KPIs definition	
Kpi_type	num	1	KPI type	Indicator for the KPIs datatype	1=char 2=num 3=date 4=time
Range_id	char	4	Range id	External key to determine the corresponding range.	0000-9999
Kpi_dep_codes	char	1024	Department codes	Codes of departments for which this KPI is applicable	String of department codes.
Kpi_subject	char	20	KPI subject	Subject determining to which perspective the KPI belongs	“financial’, “process”, “innovation” or “customer”
Kpi_direction	char	2	KPI direction	Indicator determining if a rise in the KPIs value is considered positive, negative or neutral	-1 = negative 0 = neutral 1 = positive

Table 2.1: data structure for table kpi metadata

KPI_values: table containing the actual values for the KPIs as well as the period for which this value is applicable.

Attribute	Type	Length	Label	Content	Range
Kpi_id	char	11	KPI identifier	Automatically generated, unique, internal identifier	000.000.000 – 999.999.999
Kpi_startPeriod	datetime		Start of the period	Start of the period for which the registered value is applicable	datetime
Kpi_endPeriod	datetime		End of the period	End of the period for which the registered value is applicable	datetime

Attribute	Type	Length	Label	Content	Range
Kpi_value	char	50	KPI value	Value of the KPI for the period specified	
Kpi_dep_code	char	6	Department codes	Codes of department for which this value of the KPI is applicable	Internal department codes.

Table 2.2: data structure for table kpi values

KPI_ranges: table containing the definitions of the ranges for the KPIs.

Attribute	Type	Length	Label	Content	Range
Range_id	char	4	Range id	Internal identifier for a specific range	0000-9999
Range_description	char	50	Range description	Description used for selection by the users.	
Range_startPeriod	datetime		Start of the period	Start of the period for which the range is valid	datetime
Range_endPeriod	datetime		End of the period	End of the period for which the range is valid	datetime (default 31dec2049:00:00:00.00)

Table 2.3: data structure for table kpi ranges

KPI_subranges: table containing the definitions of the subranges for specific ranges.

Attribute	Type	Length	Label	Content	Range
Subrange_id	char	6	Subrange id	Internal identifier for the specific subrange	Range_id||00-99
Range_id	char	4	Range id	External key for the linked range	0000-9999
Subrange_from	char	50	From value for the subrange	Lower limit for the subrange	
Subrange_to	char	50	To value for the subrange	Upper limit for the subrange	
Subrange_color	char	6	Subrange color	Color applicable for this subrange	Hex code defining the html-colour.

Table 2.4: data structure for table kpi subranges

KPI_departments: table containing the department codes and descriptions for selection purposes. This table is basically nothing more than a view on the organizational structure.

Attribute	Type	Length	Label	Content	Range
Kpi_dep_code	Char	6	Departmentcode	Code defining the organizational department	
Kpi_dep_description	Char	50	Department	Name of the department	

Table 2.5: data structure for table kpi departments