dunstanthomas

Friday, 17 April 2015

The Benefits of Model Driven Generator Technology (MDG)

Introduction

There are many modelling languages in international use the most common being:

- UML 2.5
- BPMN 2.0
- Archimate 2.0
- SysML 1.3

Many organisations model very successfully using these modelling languages (or a combination of them) "as-is" and they are widely supported by modelling tools such as Sparx System's Enterprise Architect.

However, it is becoming increasingly common for organisations to create their own customised version of these modelling languages (meta-models) in order to meet their modelling requirements more precisely.

Modelling tools often provide support for the creation of such modelling language customisations, and Sparx System's Enterprise Architect is at the fore-front of modelling language customisation.

Sparx Systems have named their method of customised modelling languages Model Driven Technology (MDG). This document describes what an MDG is and how its development and use can provide benefits to organisations.

What is an MDG?

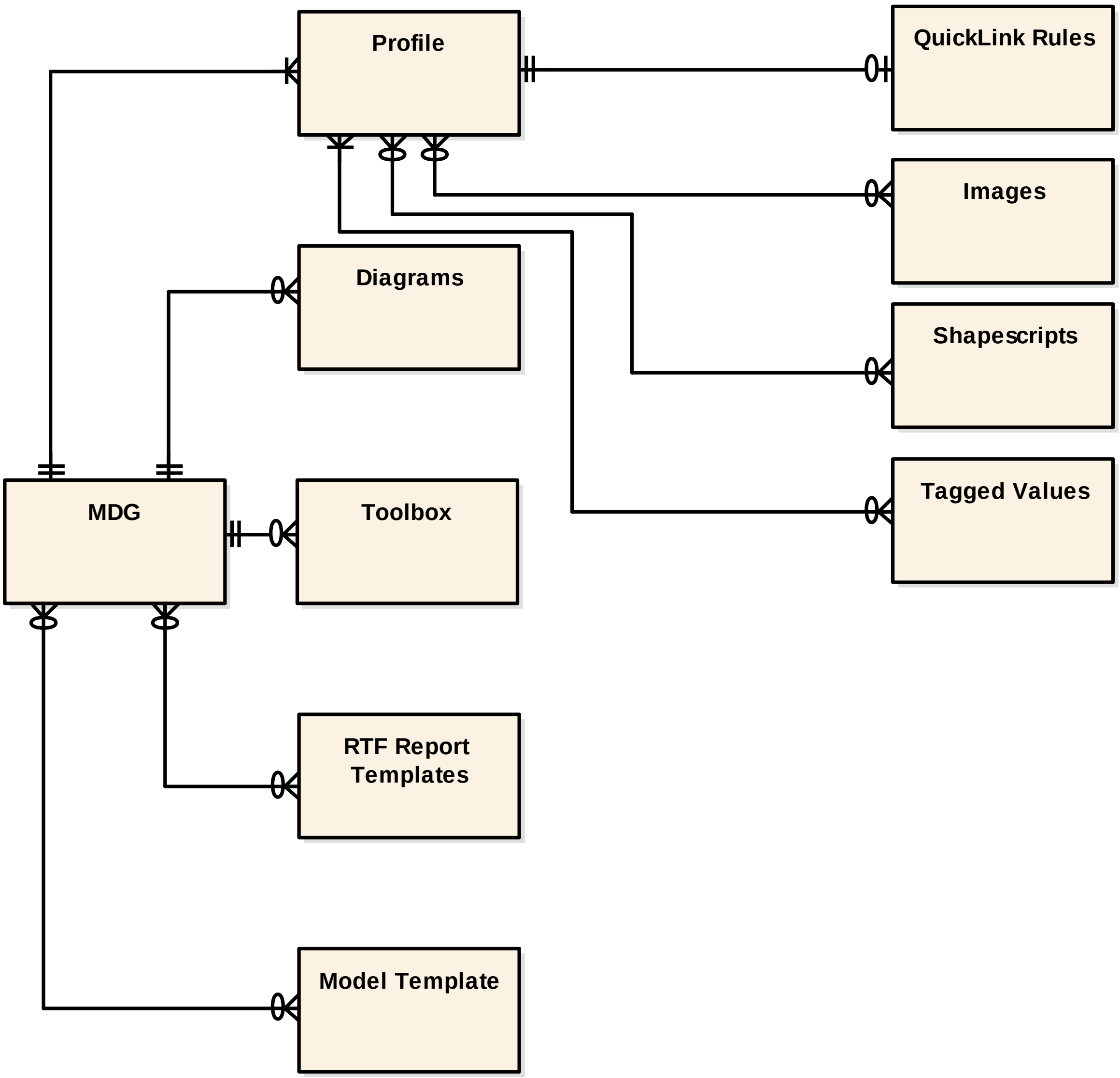
Sparx Systems define an MDG as:

"An MDG Technology is a vehicle for providing access to the resources of either a commercially-available technology or a technology that you have created yourself. Such resources include a wide range of facilities and tools, such as UML Profiles, code modules, scripts, Patterns, images, Tagged Value Types, report templates, linked document templates, Toolbox pages and Learning Centre pages.

Using Enterprise Architect, you can develop models based on the standard UML specifications, and you can extend the core UML structures using UML-supported mechanisms such as Tagged Values, Stereotypes, Profiles and Design Patterns.

These facilities are within the Enterprise Architect core technologies, and you can activate and use further Model Driven Generation (MDG) Technologies that are either integrated with the system or available from external locations." (Sparx Systems, 2014)

Diagrammatically a typical MDG is as shown below:



Added Value that an MDG Delivers to an Organisation

The benefits that an MDG brings to an organisation are as follows:

1) A standardised modelling environment that meets the modelling requirements of the organisation precisely, this includes:
    a. Symbols and colour schemes that are understood within the organisation.
    b. Data supporting the model.
    c. Reporting templates for document production.
    d. Terminology that is relevant and understood by the organisation, rather than the native modelling language "jargon".
2) A "one-stop-shop" modelling environment, where all modellers will find all modelling artefacts, (diagrams and relationships) in a single place, rather having to search the modelling tool's workspace for such artefacts.
3) Models that match precisely the meta-model, or meta-models that an organisation uses.

4) A consistent look and feel across all models with the organisation.
5) Future changes to meta-models can be applied retrospectively to existing models.
6) A single central location where the MDG is loaded automatically and hence the latest version is always available to all modellers.

These benefits have been well documented in case studies written by Sparx Systems Users from organisations ranging from domains such as defence, transportation and the financial sector. These case studies were presented at past EAUG (Enterprise Architect User Group) meetings in London, Europe and North America.

Development of an MDG

The development of a typical MDG consists of the following major phases:

1) Development of the organisation's metamodel.
2) Development of the metamodel as a profile (or profiles) within the MDG framework. This uses Enterprise Architect to model the metamodel and contains the following sub steps:
    a. Creation of tagged values to capture the data defined in the organisation metamodel.
    b. Development of Shapescripts to "draw" the shapes for elements and relationships defined in the organisations metamodel.
    c. Modelling the organisation's metamodel as a profile or profiles.
3) Development of customised toolboxes to make the profile, or profiles available to modellers.
4) Development of customised diagrams to allow modellers to present their modelling viewpoints.
5) Development of quicklinker rules for these diagrams so as to provide consistent and accurate models.
6) Developing repository structure patterns so that models are consistent across an organisation.
7) Building and testing the MDG.
8) Deploying the MDG.

The time taken for such an MDG varies according to the complexity of the organisation's metamodel, but is typically measured in man-days.

The following is an approximate estimate of man days for varying complexities of MDG:

- The simple MDG (just a customisation of elements, basic Shape scripts)
    - Element with additional fields (tagged values) and no Shapescript – 1 day per element
    - Element with additional fields (tagged values) and simple Shapescript – 2 days per element
    - Element with additional fields (tagged values) and moderate Shapescript – 3 days per element

  - Element with additional fields (tagged values) and complex Shapescript – 4 / 5 days per element (depending upon complexity of Shapescript)
- The moderate (a customised modelling environment – including model Wizard/elements (and Shapescripts)/diagrams/toolboxes/quicklinker rules) – weeks of effort
  - As for simple MDG plus the following
  - 1 day per Model Wizard
  - 1 day per customised diagram
  - 1 day per customised toolbox
  - 1 day per diagram quicklinker rules (for example if there a four diagrams, then 4 days for the diagrams + 4 days for the quicklinker rules = 8 days)
- The advanced MDG (a complete modelling environment – including model Wizard/elements (and Shapescripts)/connectors (and Shapescripts) /diagrams/toolboxes/quicklinker/patterns/resources) – months of effort
  - As for the moderate MDG plus the following
  - 1 day for the resources
  - 1 day for each model pattern

Before MDG development begins a workshop to analyse the organisations metal model would be necessary from which a more accurate estimate of man days can be obtained using the information above.

Conclusion

In order for an organisation to maximise its modelling efficiency both in terms of time and quality, the use of an MDG is becoming mandatory.

The upfront costs of the development of an MDG is quickly recovered due to the cost savings in the production of models that:

- Meet all the organisations modelling requirements precisely.
- Increase the quality and consistency of organisations model.
- Since the metamodel has been created in collaboration with stakeholders, the resultant models will meet those stakeholder requirements.
- Models are synchronised with changes in the metamodel, hence they have inbuilt longevity.

Works Cited

Sparx Systems. (2014). *Enterprise Architect User Guide.*