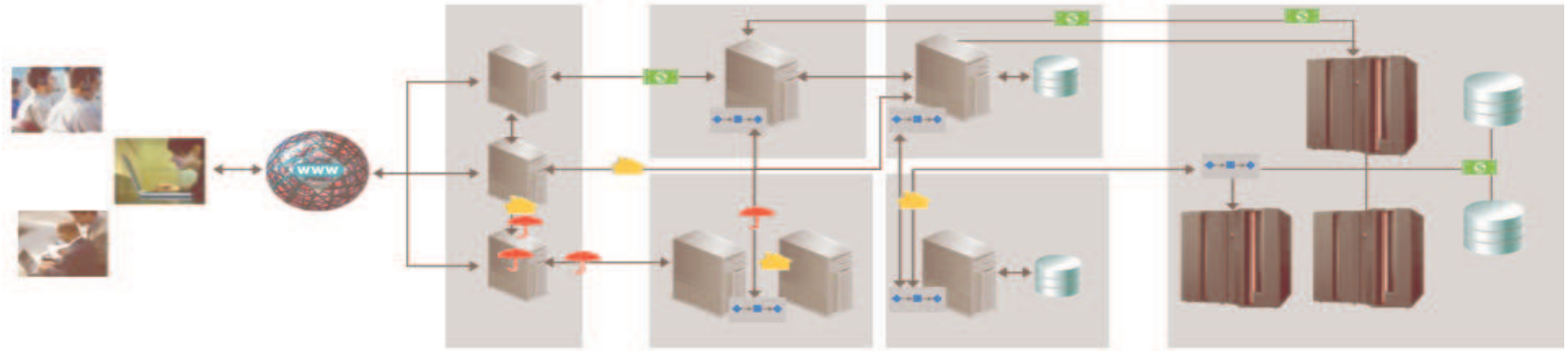
COSMOS Project Overview

DRAFT v0.5

Put together by the COSMOS community
http://wiki.eclipse.org/index.php/COSMOS

Challenges of IT Systems Management

Today's applications offer flexibility for business but introduce management challenges …



"How do we coordinate problem resolution across all parts of the organization?"

"How can we minimize the disparate information we collect?"

"How can we instrument our systems?"

"How can we share resource descriptions in a consistent manner?"

"What industry standards can be applied to help solve the problems?"

"I know something's wrong, but where?"

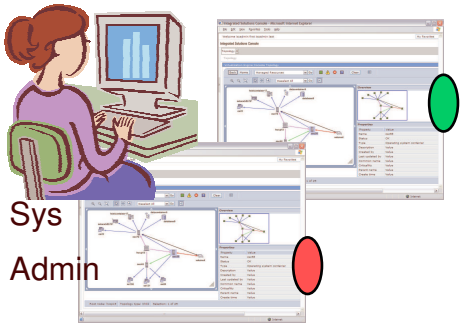
Current pain points

Each admin tool has a distinct way of representing the resources it can manage

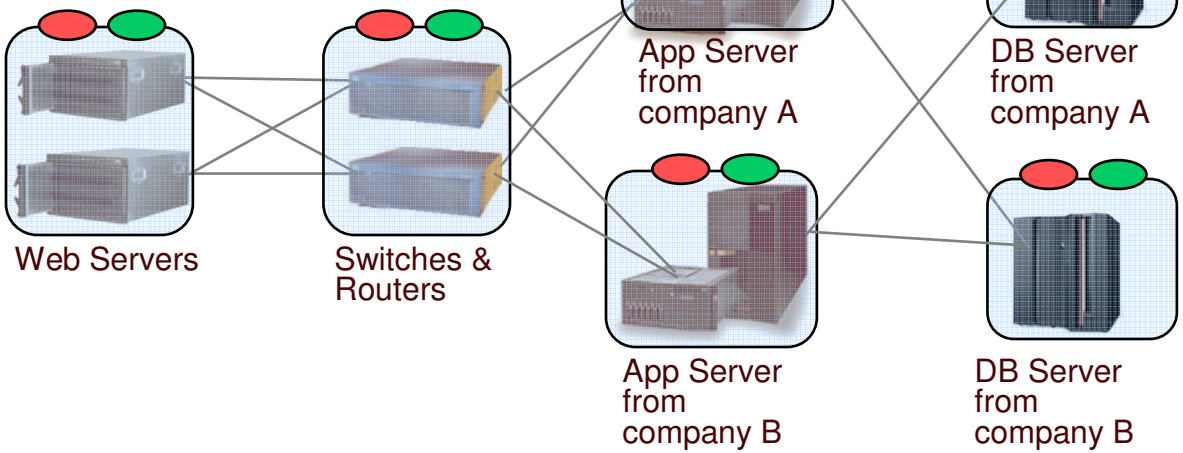
Limited interoperability b/t tools (even those from the same company)

Limited co-existence of management tools at runtime

Management tools typically targeted at a specific resource domain e.g. network fabric



Sys Admin



Customer Pain Points

- End-to-end problem determination is awkward and difficult
- Response to change in requirements is slow
- Creating higher value analytics difficult b/c of inconsistencies in the way resource information is expressed
- Inefficiencies in management costs displace development work

Often, this translates into custom instrumentation provided by management vendors.

This results in subtle, but important semantic differences in the way a resource is managed

Current pain points

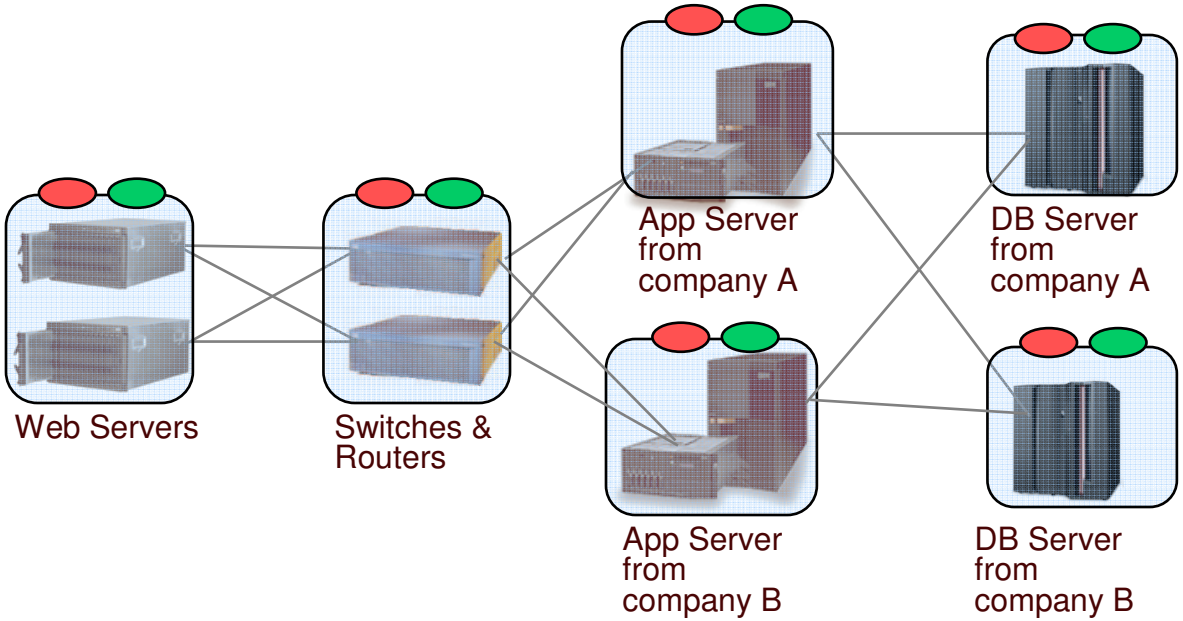
Developer

The developer has no consistent way of looking at all the resources in their environment

No consistent approach to describing the important management information

Customer Pain Points

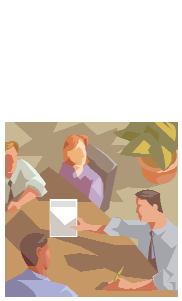
- Difficult to extend existing instrumentation tooling for new roles or management disciplines
- New “kinds” of resources difficult to add



COSMOS Subprojects

- **Resource Modeling**
- **Monitoring**
  - Data collection
  - Reporting
- **Build to Manage (BtM) a.k.a Enabling Management**

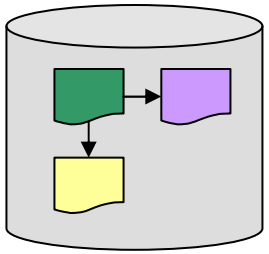
Resource Modeling Landscape



3rd Party

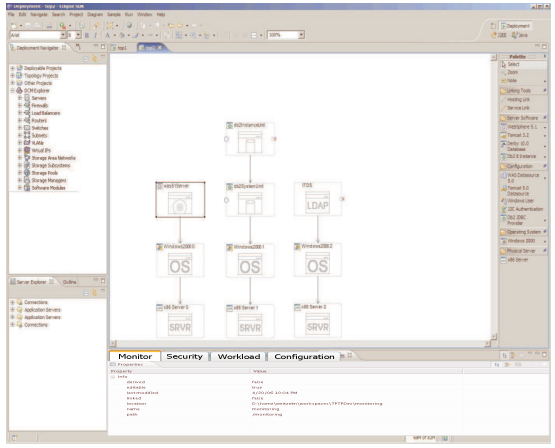
Customers

Commercial Management Vendors



Domain Specific Resource Models

Management Disciplines



- Deployment
- Monitoring
- Security
- Problem Determination
- Performance
- Availability

A domain specific model is a fixed set of genic documents and phenic templates

Each discipline specific tool deals with additional genic constraints and phenic instances

Resource Model Value

- **Value to Resource Model providers**
  - Decouples the management tools from the resource providers
- **Value to domain model consumers**
  - Decouples from resource provider
- **Value to end user**
  - Vendor neutral, extensible, tooling based upon resource models
  - Deployment and Configuration of resources
  - Simple tools to create complex models
  - Integration of Root Cause & Problem Determination

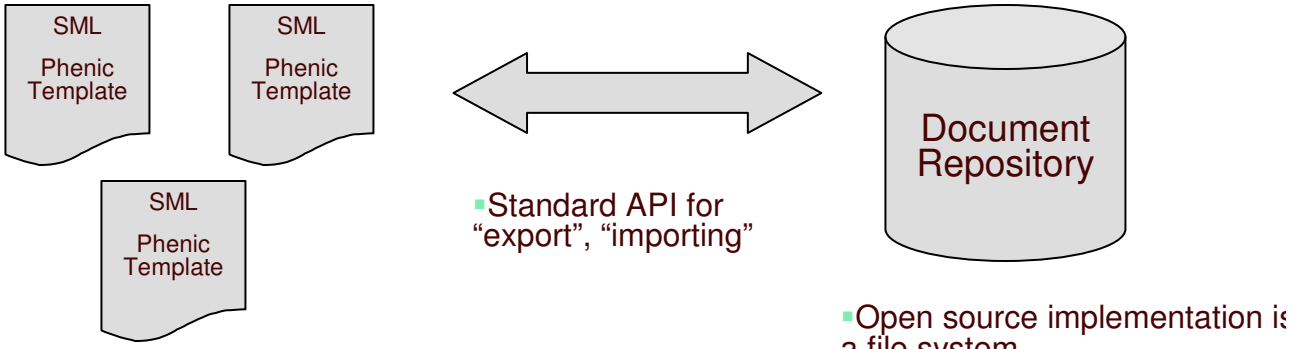
Requirements for Resource Model Ecosystem

- **Value increases exponentially with # resource models**
  - Enable creation of domain specific models
  - Demonstrate generic consumption as a building block
- **To make the models interoperable, they must be extensible**
  - Build a common repository of phenic template documents
    - Must be easy to extend
    - Easy to instantiate
  - Must provide exemplary framework to enable rapid creation of role specific tooling

COSMOS deliverables related to the creation of domain specific models

- Tooling for validating SML instances and SML-IF documents
- Tooling for importing/exporting SML-IF documents to and from predefined repositories.
  - Import and export capabilities should be extensible so that consuming products can produce implementations to deal with custom repositories
  - Open source Document Repository is implemented as a file system structure
- Tooling for creating SML template documents
  - an SML template document is an SML instance defining a common pattern that can be re-used and adapted in different domain models
- Tooling for creating domain models based on existing SML templates (Resource model builder)
  - The tool should be extendable to allow registration of third parties SML templates

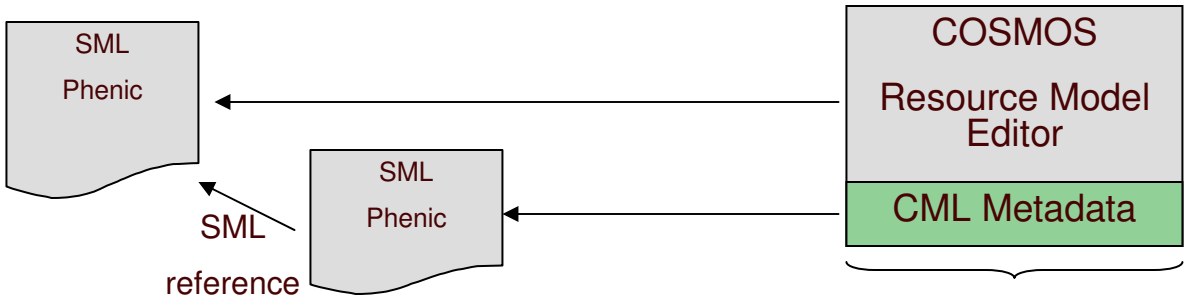
- Tooling required to create valid SML-IF documents. These may include both genic and phenic content
- "Templates" can be provided as phenic documents
- Extensible architecture to allow registration of new domain models or extensions to existing models



- Standard API for "export", "importing"
- Open source implementation is a file system

Using SML in the Monitoring life cycle: Annotating what can be observed on a resource

- Resource Model Builder
  - Based on a set of SML template document (CML metadata)
  - Extendable to use third parties template documents
  - SML template documents are used to build SML phenic instances; the result is an SML-IF document

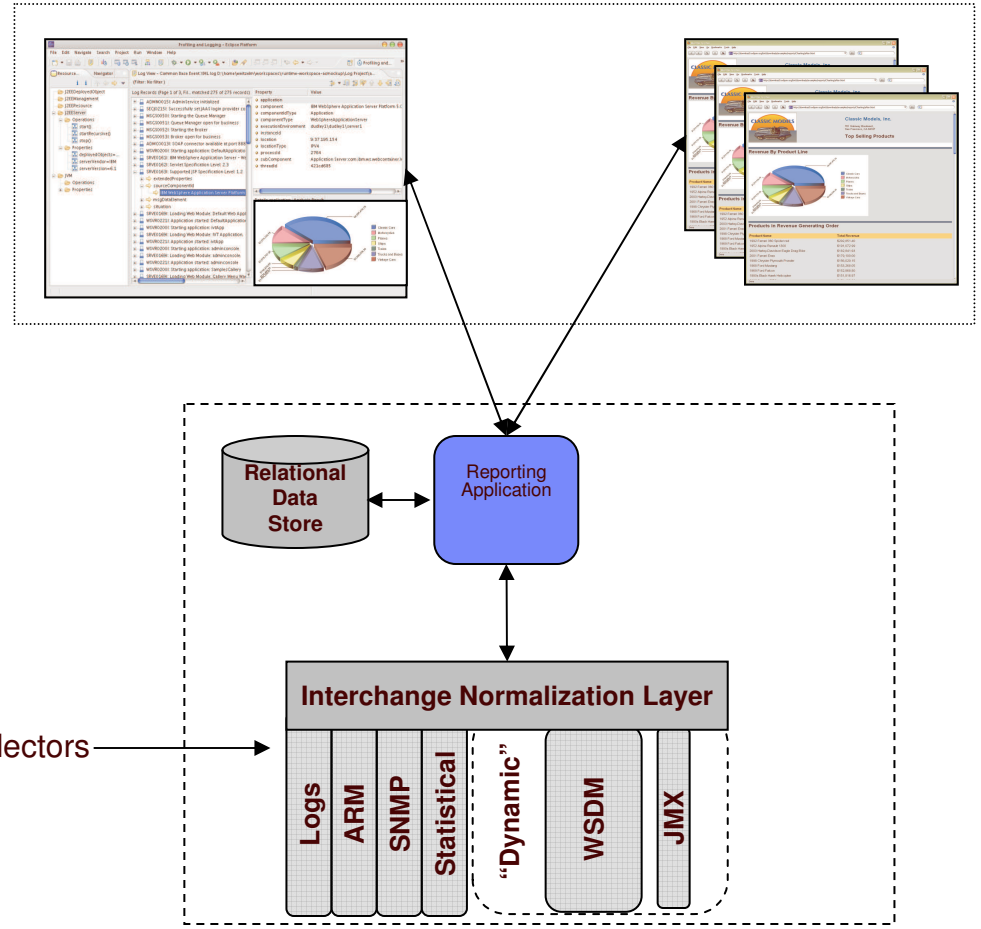


- This creates a SML phenic document that captures the monitoring metadata
- This is a CML proposal**

COSMOS Subprojects

- **Resource Modeling**
- **Build to Manage (BtM) a.k.a Enabling Management**
- **Monitoring**
  - Data collection
  - Reporting

Using BtM in the Monitoring life cycle:



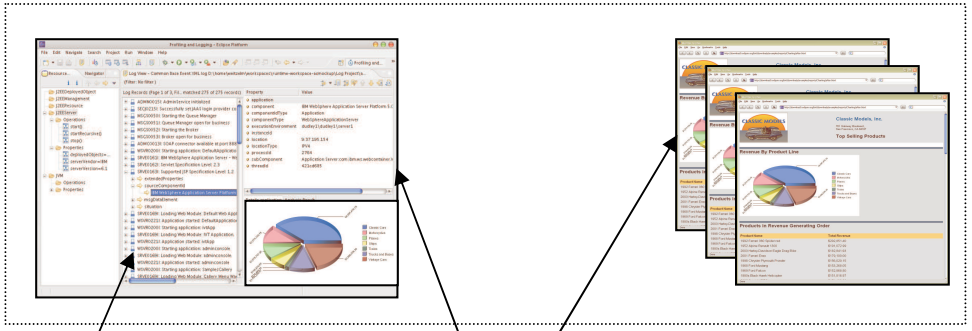
Commercial monitoring systems can:

- Reuse COSMOS data collectors

COSMOS Subprojects

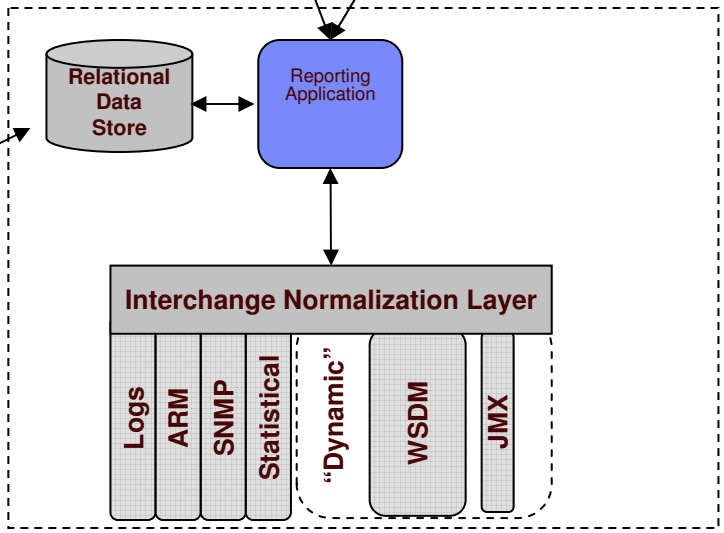
- Resource Modeling
- Build to Manage (BtM) a.k.a Enabling Management
- **Monitoring**
  - Data collection
  - Reporting

Using Data Collection & Reporting in the Monitoring life cycle:

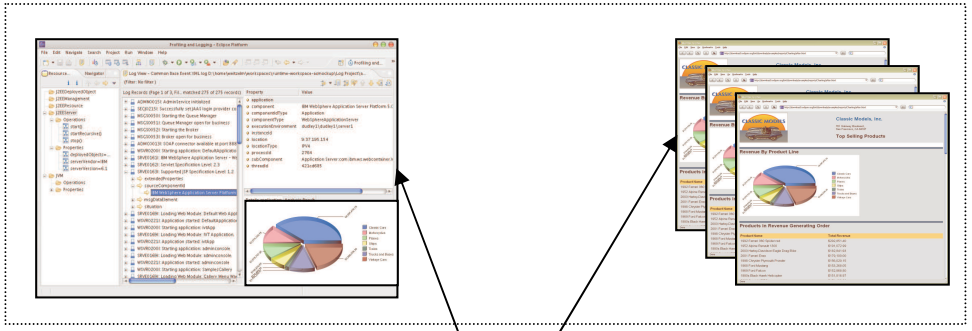


Commercial monitoring systems can:

- reuse community created reports
- extend their data by leveraging the content of the relational data store

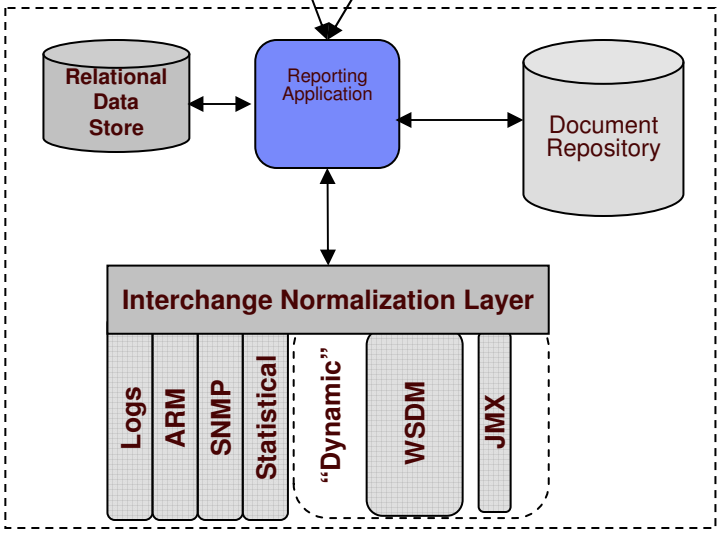


Using SML in the Monitoring life cycle: Making the observation



Commercial monitoring systems can:

- extend their data by leveraging the document repository



Use phenic documents to indicate

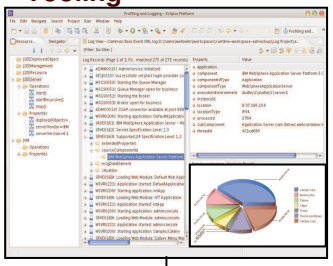
- What to collect
- What information is observable
- Phenic documents used to carry the data

COSMOS Resource monitoring and modeling tools

COSMOS is committed to providing exemplar usage of its APIs without eroding commercial opportunities

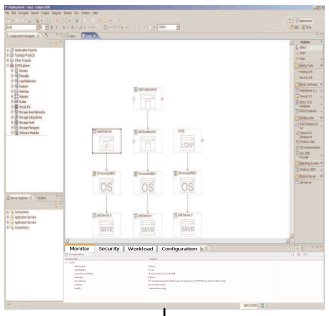
Domain specific value add plug ins can be completely in commercial space, with minimal open source capability to demonstrate and prove the framework

Build-to-Manage Tooling



Analysis | Monitoring | Problem Determination | Deployment | Security | Performance | Availability | …

Resource Model Builder



COSMOS
Monitoring
Editing / Viewing Framework based on normalized data

COSMOS
Validation/Editing/Viewing
Framework based on SML

Common framework for defining and expressing the key instrumentation points of the resource

Logs | **ARM** | **SNMP** | **Statistical** | **"Dynamic"** | **WSDM** | **JMX**

… | … | … | … | Windows | Linux | WAS | MySql | Router | JBoss | Apache

Domain specific data collectors and resource models

Commercial value add

…

Open Source

Glossary

- **SML validation**
  - The action of validating SML extensions which are provided as additions to the xml 1.0
  - Schematron validation
- **SML-IF instance validation**
  - The process of validation the content of an SML-IF resource
  - Validate SML phenic documents contained by the SML-IF resource
  - Apply any schematron rule defined as a genic document
- **Template document**
  - An SML instance defining a common pattern that can be re-used and adapted in different domain models
- **Domain models**
  - The root of an SML-IF document. Contains a set of phenic and genic documents
- **Resource domain**
  - A set of genic and/or template documents that can be used to build define a domain
- **Template editor**
  - An editor that can create genic and template documents
- **Model editor**
  - An editor that can create domain model instances, based on a set of predefined templates