MagicDraw-specific extensions

On this page

- Inspecting model decomposition
- Decomposition model
- UML model

Model manipulation describes how REST API can be used to manipulate data in Teamwork Cloud at the Eclipse modeling framework (EMF) level. However , using the API at the EMF level may become difficult when working with larger projects and more complex compositions.

Therefore, to facilitate the most common model manipulation tasks, TWC REST API has MagicDraw-specific extensions that help minimize the number of calls necessary to find the primary MagicDraw UML model within project decomposition.

Inspecting model decomposition

Normally, MagicDraw has different kinds of models within TWC resources. The term “models” here refers to the data structure required to store MagicDraw-specific project information that the tool needs when loading up projects on the client side. Currently, TWC REST API exposes two kinds of such models – **decomposition** and **UML**, as shown in the code below. You can retrieve a list of available models within the master branch and chosen resource revision by issuing GET to */resources/{resourceId}/revisions/{revision}/models*.

```
{
  "@type": [
    "ldp:Container",
    "ldp:BasicContainer"
  ],
  "ldp:contains": [
    {
      "@id": "decomposition"
    },
    {
      "@id": "uml"
    }
  ],
  "@id": "",
  "@context": {
    "ldp": "http://www.w3.org/ns/ldp#"
  }
}
```

You can further inspect the content of each of the models by issuing GET to */resources/{resourceId}/revisions/{revision}/models/{modelId}*, where *modelId* is **decomposition** or **uml** respectively.

Decomposition model

This model lists all of the Used Project hierarchy within the chosen resource. The excerpt of the results after calling */resources/{resourceId}/revisions/{revision}/models/decomposition* is shown below.

```
{
  "@type": "models:Decomposition",
  "models:usages": [
    {
      "models:project": {
        "models:name": "HSUV",
        "@type": "models:PrimaryProject",
        "@id": "../../elements/8a91f0a9-cb83-4c64-865e-e4c797556807"
      },
      "models:uses": [
        {
          "models:requiredVersion": "33",
          "models:uri": "twcloud:/45d11019-c077-4d4e-84d3-75eaf21e1230/37b7511d-bb55-4280-99b9-2da1e1cad9d6",
          "models:name": "Electronic components library",
          "@id": "../../elements/70abbe99-a551-4e46-9eda-d718c027a300"
        },
        {
          "models:requiredVersion": "19.0 v9",
          "models:uri": "file:/C:/Users/tomvil/Desktop/Cameo_Systems_Modeler_190_sp1_no_install/profiles/SysML%
20Profile.mdzip",
```

```
          "models:name": "SysML Profile",
          "@id": "../../elements/7e6d87a8-790e-4415-b1d1-493a5a2c82ac"
        }
      ]
    },
    {
      "models:project": {
        "models:name": "Time & Performance_Profile",
        "@type": "models:UsedProject",
        "@id": "../../elements/80f3fefb-77f9-49f1-afa8-6fe25951659c"
      },
      "models:uses": [
        {
          "models:requiredVersion": "19.0 v9",
          "models:uri": "file:/C:/Users/tomvil/Desktop/Cameo_Systems_Modeler_190_sp1_no_install/profiles
/UML_Standard_Profile.mdzip",
          "models:name": "UML_Standard_Profile",
          "@id": "../../elements/280a000d-ec4a-4278-aeda-15f270c99a19"
        }
      ]
    },
    {
      "models:project": {
        "models:name": "Electronic components library",
        "@type": "models:UsedProject",
        "@id": "../../elements/70abbe99-a551-4e46-9eda-d718c027a300"
      },
      "models:uses": [
        {
          "models:requiredVersion": "19.0 v9",
          "models:uri": "file:/C:/Users/tomvil/Desktop/Cameo_Systems_Modeler_190_sp1_no_install/profiles/SysML%
20Profile.mdzip",
          "models:name": "SysML Profile",
          "@id": "../../elements/7e6d87a8-790e-4415-b1d1-493a5a2c82ac"
        },
        {
          "models:requiredVersion": "19.0 v9",
          "models:uri": "file:/C:/Users/tomvil/Desktop/Cameo_Systems_Modeler_190_sp1_no_install/profiles
/UML_Standard_Profile.mdzip",
          "models:name": "UML_Standard_Profile",
          "@id": "../../elements/280a000d-ec4a-4278-aeda-15f270c99a19"
        },
        {
          "models:requiredVersion": "19.0 v9",
          "models:uri": "file:/C:/Users/tomvil/Desktop/Cameo_Systems_Modeler_190_sp1_no_install/SysML%
20constraints.mdzip",
          "models:name": "SysML constraints",
          "@id": "../../elements/fcffda41-a4a4-45d1-85bf-ee033f449701"
        }
      ]
    }
  ],
  "@id": "",
  "@context": {
    "models": "https://localhost:8111/osmc/schema/kerml/234567/models#",
    "ldp": "http://www.w3.org/ns/ldp#"
  }
}
```


The main root project which the call is executed against has a type of **models:PrimaryProject**. You can find all of its directly used projects within the **models:uses** array. Since the decomposition model covers the full hierarchy, it includes both locally used projects (usually concerning Standard/System profiles) and server projects.

Each of the directly used projects has a type of **models:UsedProject** that has its own decomposition, hence enabling the full breakdown of usage structure from the primary project perspective covering both, directly and indirectly, used projects.

⚠ The **id** attribute refers to the project ID of **esiproject:EsiProject** type, not the UML model root element.

UML model

Different from **decomposition** model previously described, the **UML** model provides a flat view of the structure of target model. Note that the term **project** should not be confused with the term **UML** model here. The purpose of the **UML** model is to expose direct links to root elements of UML models in primary projects, and both of its, directly and indirectly, used projects.

The excerpt of the results after calling */resources/{resourceId}/revisions/{revision}/models/uml* is shown below.


```
{
  "@type": "models:Uml",
  "@id": "",
  "@context": {
    "models": "https://localhost:8111/osmc/schema/kerml/234567/models#",
    "ldp": "http://www.w3.org/ns/ldp#"
  },
  "models:roots": [
    {
      "models:name": "HSUV",
      "@type": "models:PrimaryModel",
      "models:root": "../../elements/2b36804f-b904-4b29-87a5-5d438e10b696"
    },
    {
      "models:name": "Electronic components library",
      "@type": "models:UsedModel",
      "models:root": "../../elements/42dda2bc-a1cf-4ad8-bb73-88b20caa3a74"
    },
    {
      "models:name": "SysML Profile",
      "@type": "models:UsedModel",
      "models:root": "../../elements/ee2da8c9-63dc-42ec-a68a-d96dddba72df"
    },
    {
      "models:name": "UML_Standard_Profile",
      "@type": "models:UsedModel",
      "models:root": "../../elements/f325ec2b-b17c-4e59-9ddf-2e7d74e578e1"
    },
    {
      "models:name": "Fuel Supply",
      "@type": "models:UsedModel",
      "models:root": "../../elements/c4adaad7-9884-4501-a1a1-b2f515b7e6bc"
    },
    {
      "models:name": "Steering system",
      "@type": "models:UsedModel",
      "models:root": "../../elements/f1af8807-676c-4087-86ac-6e962b5bae1f"
    },
    {
      "models:name": "Time & Performance_Profile",
      "@type": "models:UsedModel",
      "models:root": "../../elements/895a1228-b850-4d8b-93b8-b33917f19952"
    }
  ]
}
```


The **models:PrimaryModel** type object provides a direct link to the main project's UML model root element via the **models:root** attribute. Compared to the EMF-based approach described in Model manipulation, using the above REST API extension is a matter of a single call and requires only the resource ID and revision number to be known in advance.

Related pages

- Developer Guide
- General convention
- Authentication
- Model manipulation