

Simplifying IOCs with EPICS Super-Modules

Zina Zhang, Márcio Paduan Donadio, Eric Gumtow

Introduction

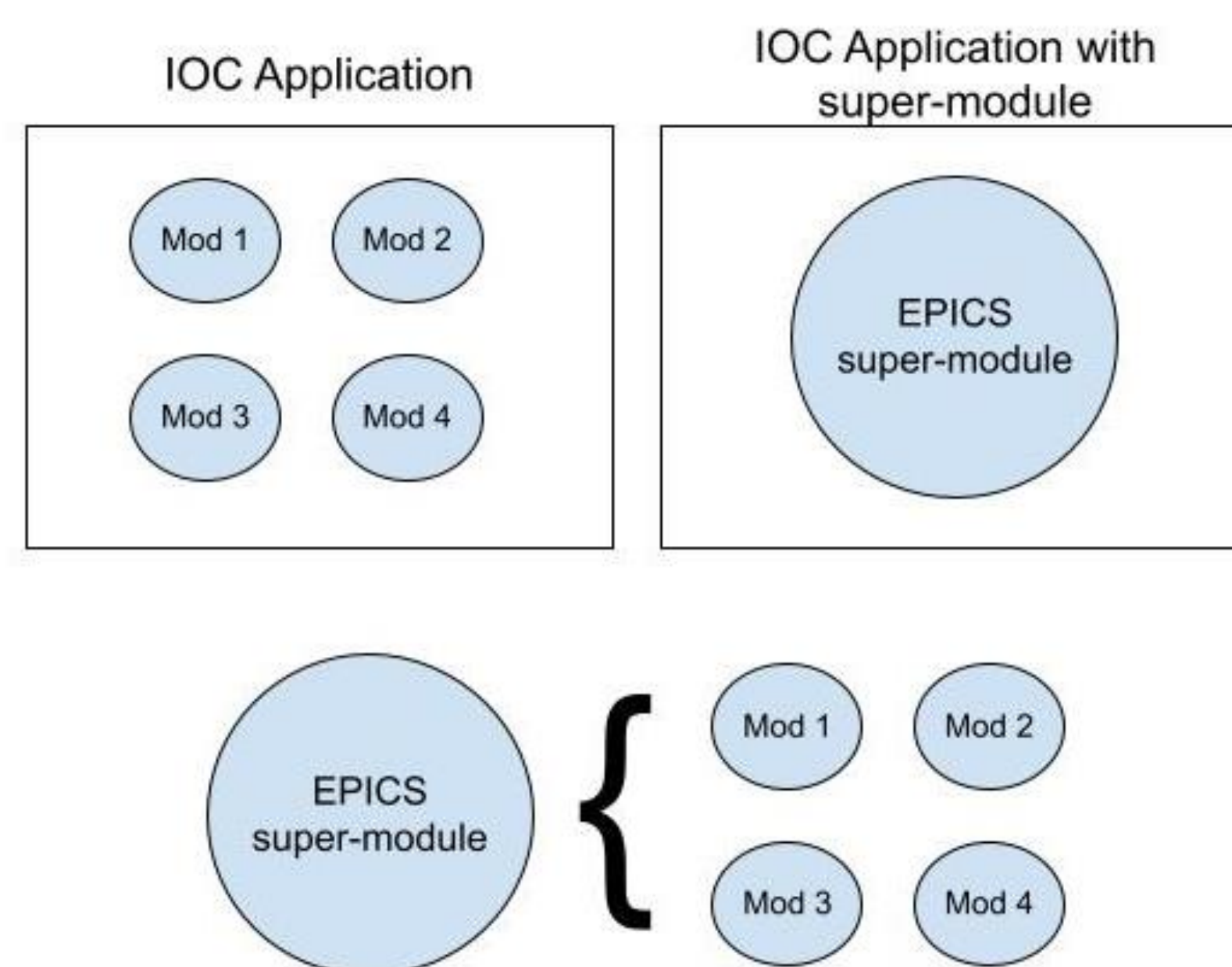
Containing collections of precompiled codes and resources, EPICS modules are a crucial part of IOC applications. Its purpose is to promote code reuse and efficiency within software development.

The Problem

- Tedious work to import a high number of modules one by one when building an application
- IOC engineers work with a list of modules. Thus, it's easier to make a mistake when there are a lot of small details
- IOC engineers must know which versions of modules to use since not all are compatible with other modules

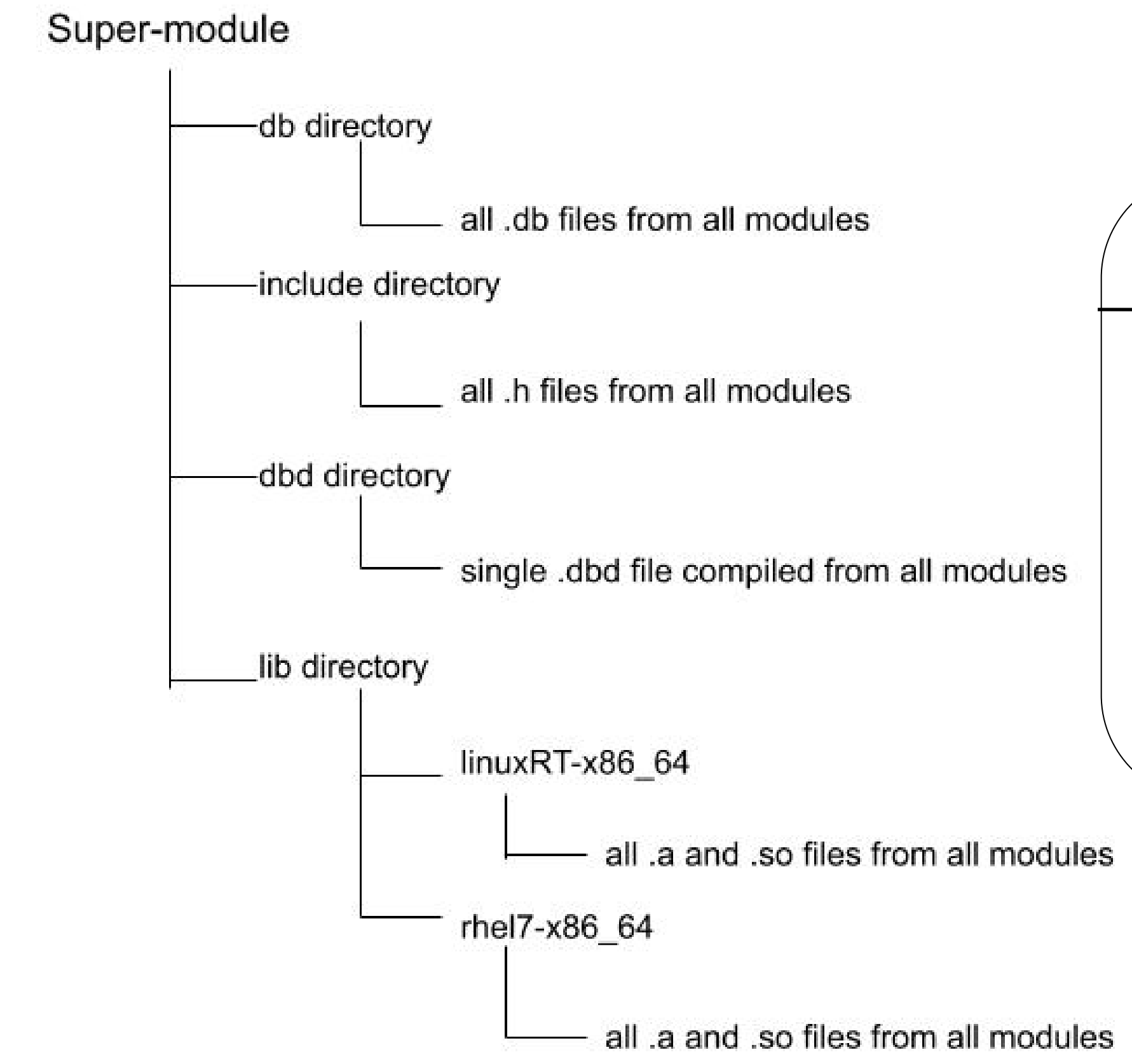
The Solution

- **Super-module**
- A giant EPICS module that encases other modules, is introduced



Process

- Create an EPICS module by following the EPICS standard
- Files worked on:
 - configure/RELEASE, configure/CONFIG_SITE, Db/Makefile, src/Makefile
- Goals:
 - Create a single .dbd file
 - Bring all .a, .so, .db, and .h files into the super-module



Note

- Make sure to test super-module on an IOC application
- Used GMD application for this step

configure/RELEASE Before

```
AUTOSAVE=$(EPICS_MODULES)/autosave/$(AUTOSAVE_MODULE_VERSION)
BLDCCLIENT=$(EPICS_MODULES)/bldClient/$(BLDCCLIENT_MODULE_VERSION)
CALC=$(EPICS_MODULES)/calc/$(CALC_MODULE_VERSION)
HISTORY=$(EPICS_MODULES)/history/$(HISTORY_MODULE_VERSION)
IOCADMIN=$(EPICS_MODULES)/iocAdmin/$(IOCADMIN_MODULE_VERSION)
CAPUTLOG=$(EPICS_MODULES)/caPutLog/$(CAPUTLOG_MODULE_VERSION)
CEXP=$(EPICS_MODULES)/cexpsh/$(CEXP_MODULE_VERSION)
PVXS=$(EPICS_MODULES)/pvxs/$(PVXS_MODULE_VERSION)
```

We don't need these modules, but CALC is not happy without them
 SNCSEQ=\$(EPICS_MODULES)/seq/R2.2.4-1.2
 SSCAN=\$(EPICS_MODULES)/sscan/\$(SSCAN_MODULE_VERSION)

```
# Needed for common platform
YAMLLOADER=$(EPICS_MODULES)/yamlLoader/$(YAMLLOADER_MODULE_VERSION)
YAMLDOWNLOADER=$(EPICS_MODULES)/yamlDownloader/$(YAMLDOWNLOADER_MODULE_VERSION)
ASYN=$(EPICS_MODULES)/asyn/$(ASYN_MODULE_VERSION)
YCPWSWASYN=$(EPICS_MODULES)/ycpswasyn/$(YCPWSWASYN_MODULE_VERSION)
ATCACOMMON=$(EPICS_MODULES)/atcaCommon/$(ATCA_COMMON_MODULE_VERSION)
TIMING=$(EPICS_MODULES)/timingApi/$(TIMING_MODULE_VERSION)
TPRPATTERN=$(EPICS_MODULES)/tprPattern/$(TPRPATTERN_MODULE_VERSION)
TPRTRIGGER=$(EPICS_MODULES)/tprTrigger/$(TPRTRIGGER_MODULE_VERSION)
BSACORE=$(EPICS_MODULES)/bsaCore/$(BSACORE_MODULE_VERSION)
BSADRIVER=$(EPICS_MODULES)/bsaDriver/$(BSADRIVER_MODULE_VERSION)
EVENT=$(EPICS_MODULES)/event/$(EVENT_MODULE_VERSION)
```

configure/RELEASE After

```
BLDCCLIENT=$(EPICS_MODULES)/bldClient/$(BLDCCLIENT_MODULE_VERSION)
CALC=$(EPICS_MODULES)/calc/$(CALC_MODULE_VERSION)
HISTORY=$(EPICS_MODULES)/history/$(HISTORY_MODULE_VERSION)
CEXP=$(EPICS_MODULES)/cexpsh/$(CEXP_MODULE_VERSION)
```

We don't need these modules, but CALC is not happy without them
 SNCSEQ=\$(EPICS_MODULES)/seq/R2.2.4-1.2
 SSCAN=\$(EPICS_MODULES)/sscan/\$(SSCAN_MODULE_VERSION)

```
EVENT=$(EPICS_MODULES)/event/$(EVENT_MODULE_VERSION)
PVXS=$(EPICS_MODULES)/pvxs/$(PVXS_MODULE_VERSION)
```

```
COMMON_MODS=/u/gu/zhang12/epics/iocCommonSupportLib
ATCACOMPLETE_MODS=/u/gu/zhang12/epics/atcaCompleteBundle
```

- Number of modules/packages is drastically reduced
- Reduces chances of mistakes being made

configure/CONFIG_SITE Before

```
CPFW_DEVICE_LIBRARY_PACKAGE_NAME=cpsw/deviceLibrary
CPFW_DEVICE_LIBRARY_VERSION=1.2.0
CPFW_DEVICE_LIBRARY_PACKAGE_SITE_TOP=$(CPFW_DEVICE_LIBRARY_PACKAGE_NAME)/$(CPFW_DEVICE_LIBRARY_VERSION)
CPFW_DEVICE_LIBRARY_LIB=$(CPFW_DEVICE_LIBRARY_PACKAGE_SITE_TOP)/$(PKG_ARCH)/lib
CPFW_DEVICE_LIBRARY_INCLUDE=$(CPFW_DEVICE_LIBRARY_PACKAGE_SITE_TOP)/$(PKG_ARCH)/include
```

We don't need these modules, but CALC is not happy without them
 SNCSEQ=\$(EPICS_MODULES)/seq/R2.2.4-1.2
 SSCAN=\$(EPICS_MODULES)/sscan/\$(SSCAN_MODULE_VERSION)

```
# Needed for common platform
YAMLLOADER=$(EPICS_MODULES)/yamlLoader/$(YAMLLOADER_MODULE_VERSION)
YAMLDOWNLOADER=$(EPICS_MODULES)/yamlDownloader/$(YAMLDOWNLOADER_MODULE_VERSION)
ASYN=$(EPICS_MODULES)/asyn/$(ASYN_MODULE_VERSION)
YCPWSWASYN=$(EPICS_MODULES)/ycpswasyn/$(YCPWSWASYN_MODULE_VERSION)
ATCACOMMON=$(EPICS_MODULES)/atcaCommon/$(ATCA_COMMON_MODULE_VERSION)
TIMING=$(EPICS_MODULES)/timingApi/$(TIMING_MODULE_VERSION)
TPRPATTERN=$(EPICS_MODULES)/tprPattern/$(TPRPATTERN_MODULE_VERSION)
TPRTRIGGER=$(EPICS_MODULES)/tprTrigger/$(TPRTRIGGER_MODULE_VERSION)
BSACORE=$(EPICS_MODULES)/bsaCore/$(BSACORE_MODULE_VERSION)
BSADRIVER=$(EPICS_MODULES)/bsaDriver/$(BSADRIVER_MODULE_VERSION)
EVENT=$(EPICS_MODULES)/event/$(EVENT_MODULE_VERSION)
```

configure/CONFIG_SITE After

```
BLDCCLIENT=$(EPICS_MODULES)/bldClient/$(BLDCCLIENT_MODULE_VERSION)
CALC=$(EPICS_MODULES)/calc/$(CALC_MODULE_VERSION)
HISTORY=$(EPICS_MODULES)/history/$(HISTORY_MODULE_VERSION)
CEXP=$(EPICS_MODULES)/cexpsh/$(CEXP_MODULE_VERSION)
```

We don't need these modules, but CALC is not happy without them
 SNCSEQ=\$(EPICS_MODULES)/seq/R2.2.4-1.2
 SSCAN=\$(EPICS_MODULES)/sscan/\$(SSCAN_MODULE_VERSION)

```
EVENT=$(EPICS_MODULES)/event/$(EVENT_MODULE_VERSION)
PVXS=$(EPICS_MODULES)/pvxs/$(PVXS_MODULE_VERSION)
```

```
COMMON_MODS=/u/gu/zhang12/epics/iocCommonSupportLib
ATCACOMPLETE_MODS=/u/gu/zhang12/epics/atcaCompleteBundle
```

- GMD application built successfully

Results

- Created two main super-modules: iocCommonSupportLib and atcaCompleteBundle (composed of atcaNcBundle & atcaScBundle)
- Modules in iocCommonSupportLib:
 - autosave, iocAdmin, caPutLog
- Modules in complete bundle:
 - asyn, bsaDriver, yamlLoader, yamlDownloader, tprTrigger, tprPattern, ycpswasyn, timingapi, BsaCore, ATCACommon
- Packages in complete bundle:
 - cpsw/framework, boost, yaml-cpp, cpsw/yamlReader, cpsw/deviceLibrary, timing/hpsTpr, timing/bsa, atca/commonATCA
- event module couldn't be added to the super-module due to function names duplicating

Next Steps

- Combine all .a files within each architecture into a single .a file
- 'ar' command will be used
- Must create a new RULES file that'll make it automatic

Acknowledgments

I would like to thank my mentor, Márcio Paduan Donadio, and Eric Gumtow for their guidance and support throughout this project.