

g-2 Simulation Software Workshop Feb 2014

INTRODUCTION

Adam Lyon/FNAL SCD

What's in it for me?

What's in it for me?

Learn:

What's in it for me?

Learn:

Why we should switch from what we have now

What's in it for me?

Learn:

Why we should switch from what we have now

What's different about the Art software system

What's in it for me?

Learn:

Why we should switch from what we have now

What's different about the Art software system

How to program in Art & ArtG4

What's in it for me?

Learn:

Why we should switch from what we have now

What's different about the Art software system

How to program in Art & ArtG4

Some best software practices

What's in it for me?

Learn:

Why we should switch from what we have now

What's different about the Art software system

How to program in Art & ArtG4

Some best software practices

How to work easily together

What's in it for me (Adam)?

What's in it for me (Adam)?

I need your help!

What's in it for me (Adam)?

I need your help!

I go to too many meetings to do this all myself

What's in it for me (Adam)?

I need your help!

I go to too many meetings to do this all myself

I want you up to speed so you can contribute

What's in it for me (Adam)?

I need your help!

I go to too many meetings to do this all myself

I want you up to speed so you can contribute

I want to work together with you easily

Working together easily?

Working together easily?

E.g. Tasha Arvanitis was a g-2 summer student who wrote much of ArtG4. Her help was awesome.

Working together easily?

E.g. Tasha Arvanitis was a g-2 summer student who wrote much of ArtG4. Her help was awesome.

She wasn't a C++ guru nor a Geant guru.

Working together easily?

E.g. Tasha Arvanitis was a g-2 summer student who wrote much of ArtG4. Her help was awesome.

She wasn't a C++ guru nor a Geant guru.

I think the *system* made it easy for her to make a big contribution (plus she was super-smart).

Working together easily?

E.g. Tasha Arvanitis was a g-2 summer student who wrote much of ArtG4. Her help was awesome.

She wasn't a C++ guru nor a Geant guru.

I think the *system* made it easy for her to make a big contribution (plus she was super-smart).

The idea is that with Art, you can easily make a big contribution too, even if you aren't a C++ guru and non-super-smart like me

**But you need to work in a way
that makes it easy to work with
others**

Making it easy to work together?

Making it easy to work together?

What does this mean?

Making it easy to work together?

What does this mean?

Following best coding practices?

Making it easy to work together?

What does this mean?

Following best coding practices?

Using standard libraries and APIs?

Making it easy to work together?

What does this mean?

Following best coding practices?

Using standard libraries and APIs?

Creating your own libraries for others to use?

Making it easy to work together?

What does this mean?

Following best coding practices?

Using standard libraries and APIs?

Creating your own libraries for others to use?

Share your code in a repository?

Making it easy to work together?

What does this mean?

Following best coding practices?

Using standard libraries and APIs?

Creating your own libraries for others to use?

Share your code in a repository?

Documenting your code?

Making it easy to work together?

What does this mean?

Following best coding practices?

Using standard libraries and APIs?

Creating your own libraries for others to use?

Share your code in a repository?

Documenting your code?

Yes to all of the above!

How to work easily together

How to work easily together

A software system and framework can make this possible

How to work easily together

A software system and framework can make this possible

Some people find such a system constraining:

How to work easily together

A software system and framework can make this possible

**Some people find such a system constraining:
Infrastructure is hidden behind the scenes from you**

How to work easily together

A software system and framework can make this possible

Some people find such a system constraining:
Infrastructure is hidden behind the scenes from you
Your ideas may not be included

How to work easily together

A software system and framework can make this possible

Some people find such a system constraining:

Infrastructure is hidden behind the scenes from you

Your ideas may not be included

You have to trust a system you didn't write

How to work easily together

A software system and framework can make this possible

Some people find such a system constraining:
Infrastructure is hidden behind the scenes from you
Your ideas may not be included
You have to trust a system you didn't write

Some people find such a system liberating:

How to work easily together

A software system and framework can make this possible

Some people find such a system constraining:

Infrastructure is hidden behind the scenes from you

Your ideas may not be included

You have to trust a system you didn't write

Some people find such a system liberating:

You can concentrate on physics code

How to work easily together

A software system and framework can make this possible

Some people find such a system constraining:

Infrastructure is hidden behind the scenes from you

Your ideas may not be included

You have to trust a system you didn't write

Some people find such a system liberating:

You can concentrate on physics code

You can use code from others and share yours

How to work easily together

A software system and framework can make this possible

Some people find such a system constraining:

Infrastructure is hidden behind the scenes from you

Your ideas may not be included

You have to trust a system you didn't write

Some people find such a system liberating:

You can concentrate on physics code

You can use code from others and share yours

Things might “just work”

Using a system from others

Using a system from others

I don't think we have much of a choice

Using a system from others

I don't think we have much of a choice

Unlike Run II and LHC, we don't have the time, the expertise and the FTEs to create a system ourselves

Using a system from others

I don't think we have much of a choice

Unlike Run II and LHC, we don't have the time, the expertise and the FTEs to create a system ourselves

Though designing and writing a system is fun, there are so many devils in the details

Using a system from others

I don't think we have much of a choice

Unlike Run II and LHC, we don't have the time, the expertise and the FTEs to create a system ourselves

Though designing and writing a system is fun, there are so many devils in the details

Like the rest of IF, we are relying on the ART developers to do much of the infrastructure work for us

Why must we make it easy to work together?

Why must we make it easy to work together?

The Real Answer:

Why must we make it easy to work together?

The Real Answer:

Science demands reproducibility

Why must we make it easy to work together?

The Real Answer:

Science demands reproducibility

Results must come from official code

Why must we make it easy to work together?

The Real Answer:

Science demands reproducibility

Results must come from official code

We must be able to write our experiment software and maintain our sanity

Why must we make it easy to work together?

The Real Answer:

Science demands reproducibility

Results must come from official code

We must be able to write our experiment software and maintain our sanity

Fermilab IF experiments believe that ART is on the right path

Are we giving it all away?

Are we giving it all away?

NO! Our voices count.

Are we giving it all away?

NO! Our voices count.

ART Stakeholders meetings are every Friday

Are we giving it all away?

NO! Our voices count.

ART Stakeholders meetings are every Friday

We still need to do validation

Are we giving it all away?

NO! Our voices count.

ART Stakeholders meetings are every Friday

We still need to do validation

The stuff I've written is not set in stone

Are we giving it all away?

NO! Our voices count.

ART Stakeholders meetings are every Friday

We still need to do validation

The stuff I've written is not set in stone

Complain early and often

Are we giving it all away?

NO! Our voices count.

ART Stakeholders meetings are every Friday

We still need to do validation

The stuff I've written is not set in stone

Complain early and often

But be ready to help (and compromise)

**Why should we not stay with
what we have (e.g. g2migrace)?**

**Why should we not stay with
what we have (e.g. g2migrace)?**

Because it isn't a framework

Why should we not stay with what we have (e.g. g2migrace)?

Because it isn't a framework

It's not modular

Why should we not stay with what we have (e.g. g2migrace)?

Because it isn't a framework

It's not modular

It's hard to add new functionality in a “work together easily way”

Why should we not stay with what we have (e.g. g2migrace)?

Because it isn't a framework

It's not modular

It's hard to add new functionality in a “work together easily way”

Case in point...

In g2migtrace/src/primaryConstruction.cc

```
// constructionMaterials is essentially a "materials library" class.
// Passing to to construction functions allows access to all materials

/**** BEGIN CONSTRUCTION PROCESS ****/

// Construct the world volume
labPTR = lab -> ConstructLab();
// Construct the "holders" of the actual physical objects
#ifdef TESTBEAM
    Arch.push_back(labPTR);
#else
    Arch = arc->ConstructArcs(labPTR);
#endif
// Build the calorimeters
// cal -> ConstructCalorimeters(Arch);
// station->ConstructStations(Arch);
#ifdef TESTBEAM
// Build the physical vacuum chambers and the vacuum itself
Vach = vC -> ConstructVacChamber(Arch);
```

In g2migtrace/src/primaryConstruction.cc

```
// constructionMaterials is essentially a "materials library" class.
// Passing to to construction functions allows access to all materials

/**** BEGIN CONSTRUCTION PROCESS ****/

// Construct the world volume
labPTR = lab -> ConstructLab();
// Construct the "holders" of the actual physical objects
#ifdef TESTBEAM
    Arch.push_back(labPTR);
#else
    Arch = arc->ConstructArcs(labPTR);
#endif
// Build the calorimeters
// cal -> ConstructCalorimeters(Arch);
// station->ConstructStations(Arch);
#ifdef TESTBEAM
// Build the physical vacuum chambers and the vacuum itself
Vach = vC -> ConstructVacChamber(Arch);
```

We can't simultaneously maintain this code and our sanity

In g2migtrace/src/primaryConstruction.cc

```
// constructionMaterials is essentially a "materials library" class.  
// Passing to to construction functions allows access to all materials
```

```
/*  
**** BEGIN CONSTRUCTION PROCESS *****/
```

WHAT IF WE HAVE A
DIFFERENT TEST BEAM?

```
// Construct the world volume
```

```
labPTR = lab -> ConstructLab();
```

```
// Construct the "holders" of the actual physical objects
```

```
#ifdef TESTBEAM
```

```
Arch.push_back(labPTR);
```

```
#else
```

```
Arch = arc->ConstructArcs(labPTR);
```

```
#endif
```

```
// Build the calorimeters
```

```
// cal -> ConstructCalorimeters(Arch);
```

```
station->ConstructStations(Arch);
```

THIS KIND OF CODE IS
HARD TO EXCISE LATER

```
#ifndef TESTBEAM
```

```
// Build the physical vacuum chambers and the vacuum itself
```

```
Vach = vC -> ConstructVacChamber(Arch);
```

**We can't simultaneously maintain this
code and our sanity**

Maintaining sanity is hard

Maintaining sanity is hard

It's not your fault - you just want results!

Maintaining sanity is hard

It's not your fault - you just want results!

We don't have a system that tries to make things easy

Maintaining sanity is hard

It's not your fault - you just want results!

We don't have a system that tries to make things easy

It's not the system's fault - it wasn't written for that

Maintaining sanity is hard

It's not your fault - you just want results!

We don't have a system that tries to make things easy

It's not the system's fault - it wasn't written for that

Writing such a system is hard (need experts)

Maintaining sanity is hard

It's not your fault - you just want results!

We don't have a system that tries to make things easy

It's not the system's fault - it wasn't written for that

Writing such a system is hard (need experts)

Learning such a system is non-trivial. That's why you are here.

What's does a system need?

What's does a system need?

<1> Source version control and a place for documentation

What's does a system need?

<1> Source version control and a place for documentation

<2> A build system

What's does a system need?

<1> Source version control and a place for documentation

<2> A build system

<3> A Release, Dependency, and Environment management system

What's does a system need?

<1> Source version control and a place for documentation

<2> A build system

<3> A Release, Dependency, and Environment management system

<4> A software framework

What's does a system need?

<1> Source version control and a place for documentation

<2> A build system

<3> A Release, Dependency, and Environment management system

<4> A software framework

I've tried to follow the Art developers and their choices, but hopefully with some improvements

<1> Source code version control

<1> Source code version control

Redmine & git

<1> Source code version control

Redmine & git

git-flow for easier branch “encouragement”

<1> Code documentation

<1> Code documentation

Codedocify + Redmine

<1> Code documentation

Codedocify + Redmine

Codedocify: Keep the documentation with the code for easy updating

<1> Code documentation

Codedocify + Redmine

Codedocify: Keep the documentation with the code for easy updating

But display nicely

<2> Build system

<2> Build system

How do you specify what to build?

<2> Build system

How do you specify what to build?

How are the build “products” organized?

<2> Build system

How do you specify what to build?

How are the build “products” organized?

How do you go from builds to running?

<2> Build system

How do you specify what to build?

How are the build “products” organized?

How do you go from builds to running?

CMake with mrb wrapper

<3> Release, Dependency & Environment management

<3> Release, Dependency & Environment management

What versions of code and libraries go together?

<3> Release, Dependency & Environment management

What versions of code and libraries go together?

Handle different “flavors” (e.g. SLF5/6, debug/opt)

<3> Release, Dependency & Environment management

What versions of code and libraries go together?

Handle different “flavors” (e.g. SLF5/6, debug/opt)

Manage an environment

<3> Release, Dependency & Environment management

What versions of code and libraries go together?

Handle different “flavors” (e.g. SLF5/6, debug/opt)

Manage an environment

Ensure a consistent build

Consistent builds?

Consistent builds?

All code compiled with the same compiler, same compiler options, and same flavor

Consistent builds?

All code compiled with the same compiler, same compiler options, and same flavor

Inconsistent builds lead to horribly subtle and complicated problems. We can waste weeks trying to figure them out

Consistent builds?

Consistent builds?

Case in point: I was trying to build g2MIGTRACE with C++11 compiler options turned on. Resulting executable seg-faulted in Root library loading!!

Consistent builds?

Case in point: I was trying to build g2MIGTRACE with C++11 compiler options turned on. Resulting executable seg-faulted in Root library loading!!

After THREE WEEKS of me and experts debugging, finally determined that the geant build system had a bug which allowed some geant libraries to be built without C++11 compiler options

Consistent builds?

Case in point: I was trying to build g2MIGTRACE with C++11 compiler options turned on. Resulting executable seg-faulted in Root library loading!!

After THREE WEEKS of me and experts debugging, finally determined that the geant build system had a bug which allowed some geant libraries to be built without C++11 compiler options

This was figured out by debugging ASSEMBLY CODE

Consistent builds?

Case in point: I was trying to build g2MIGTRACE with C++11 compiler options turned on. Resulting executable seg-faulted in Root library loading!!

After THREE WEEKS of me and experts debugging, finally determined that the geant build system had a bug which allowed some geant libraries to be built without C++11 compiler options

This was figured out by debugging ASSEMBLY CODE

We don't want to do this. It's horrible. Consistent builds are crucial

<3> Release, dependency and environment management

<3> Release, dependency and environment management

Use “Relocatable UPS” (e.g. setup)

Same as the Art developers use

Droppable tar files can be distributed

<3> Release, dependency and environment management

Use “Relocatable UPS” (e.g. setup)

Same as the Art developers use

Droppable tar files can be distributed

Wrapped in “mrb” scripts to make procedures and tasks easy

**Use “Relocatable UPS” (e.g. setup)
Same as the Art developers use
Droppable tar files can be distributed**

Use “Relocatable UPS” (e.g. setup)

Same as the Art developers use

Droppable tar files can be distributed

Wrapped in “mrb**” scripts to make procedures and tasks easy**

mr**b**

Use “Relocatable UPS” (e.g. setup)

Same as the Art developers use

Droppable tar files can be distributed

Wrapped in “mrb**” scripts to make procedures and tasks easy**

mrb** newDev**

mrb** gitCheckout**

Use “Relocatable UPS” (e.g. setup)

Same as the Art developers use

Droppable tar files can be distributed

Wrapped in “mrb**” scripts to make procedures and tasks easy**

mrb** newDev**

mrb** newProduct**

mrb** gitCheckout**

mrb** zapBuild**

Use “Relocatable UPS” (e.g. setup)

Same as the Art developers use

Droppable tar files can be distributed

Wrapped in “mrB” scripts to make procedures and tasks easy

mrB newDev

mrB newProduct

source mrB setEnv

mrB gitCheckout

mrB zapBuild

Use “Relocatable UPS” (e.g. setup)

Same as the Art developers use

Droppable tar files can be distributed

Wrapped in “mrb**” scripts to make procedures and tasks easy**

mrb** newDev**

mrb** newProduct**

source mrb** setEnv**

mrb** build**

mrb** gitCheckout**

mrb** zapBuild**

gm2d

Constraining and liberating:

Constraining and liberating:

L: Shields you somewhat from a complicated underlying system that may change (git, ups, cmake)

Constraining and liberating:

L: Shields you somewhat from a complicated underlying system that may change (git, ups, cmake)

C: You don't know how things really work

Constraining and liberating:

L: Shields you somewhat from a complicated underlying system that may change (git, ups, cmake)

C: You don't know how things really work

**But these are just scripts. See
\$MRB_DIR/bin**

<4> A software framework

<4> A software framework

Modular

<4> A software framework

Modular

Handles i/o, library loading, persistency, services

<4> A software framework

Modular

Handles i/o, library loading, persistency, services

Best practices C++

<4> A software framework

Modular

Handles i/o, library loading, persistency, services

Best practices C++

You write your piece of the puzzle

<4> A software framework

Modular

Handles i/o, library loading, persistency, services

Best practices C++

You write your piece of the puzzle

You don't have to build the table (and the floor, and the house, and the foundation, ...)

Art provides a data format

Art provides a data format

Right now it is Root based

Art provides a data format

Right now it is Root based

Could be something else, but would take a lot of work

Art provides a data format

Right now it is Root based

Could be something else, but would take a lot of work

Writing a translator from, say, MIDAS would probably not be too bad

Art provides a data format

Right now it is Root based

Could be something else, but would take a lot of work

Writing a translator from, say, MIDAS would probably not be too bad

You do your usual thing of writing histograms and Root trees

Art provides a data format

Right now it is Root based

Could be something else, but would take a lot of work

Writing a translator from, say, MIDAS would probably not be too bad

You do your usual thing of writing histograms and Root trees

But try to do as much of your analysis in Art as possible

Art provides a data format

Right now it is Root based

Could be something else, but would take a lot of work

Writing a translator from, say, MIDAS would probably not be too bad

You do your usual thing of writing histograms and Root trees

But try to do as much of your analysis in Art as possible

Let's try not to develop an ecosystem based on Root trees