

Containers in CCA

What's here, What's possible, What we need to do

April 24, 2008

Benjamin Allan

baallan@sandia.gov





Caveats:

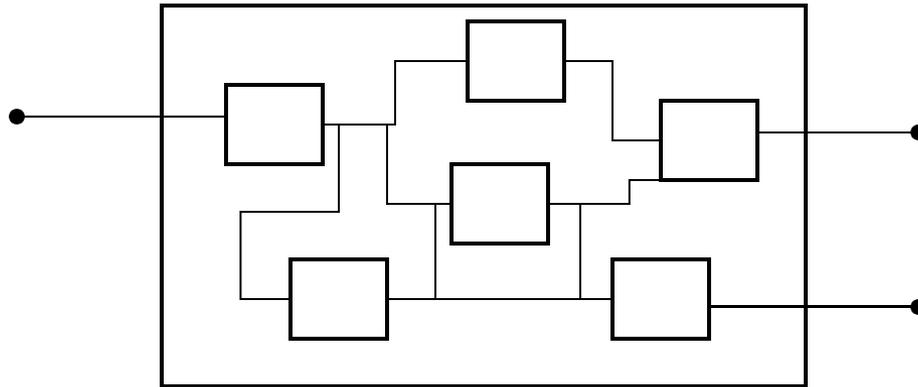
- ❑ Discussion limited to in-process (possibly spmd, mpmd).
- ❑ Extension to RMI apps is not conceptually different, but...
 - The details could be ugly, for the hierarchical approaches.





Container: group of components with details *hidden*

- Means: Pre-wire a set of components and export unwired ports to the *visible* world as if the set was a single entity.



- Aim: simplify creation and understanding of complex applications.





What's here

- ❑ Multiple source code routes to creation of applications from subassemblies that appear as components.
- ❑ Existing GUI(s?) *do not* support any of those routes.
- ❑ In-progress is a converter from application assembly script saved by ccafe-gui to stand-alone main() that is easily (but manually in the near term) converted to a component.





Q: What is the needed/desired approach?

- ❑ Several are available.
- ❑ Resources are limited— prioritize please, then volunteer.
- ❑ All require significant GUI development to make them genuinely end-user friendly.
- ❑ All are attractive targets for additional code generation tools to hide BuilderService details from graphically oriented users.
- ❑ Just what/kind how much of hiding do we *really* want, given that this is scientific programming and we ultimately need total control of parameter tuning?

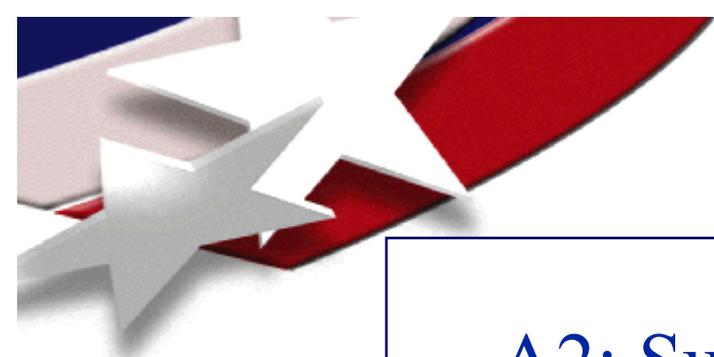




A1: Cosmetics (virtual components)

- ❑ One-frame (flat component universe).
- ❑ GUI responsible for groupings and visibilities on a per user basis. Components/connections dynamically rendered. Think (Zoom+/Zoom-). *Don't think ccafe-gui.*
- ❑ Most powerful approach for end user, with right GUI.
- ❑ Multiple simultaneous views possible for large applications with distributed users.
- ❑ Canned subassemblies created via GUI scripts.
- ❑ No specification extensions required.





A2: Subassembly by a manager X

- ❑ One component, X, creates and connects a set of others ([Xsubs]), re-exports unconnected ports as its own.
- ❑ User needs to wire up X. No knowledge of [Xsubs].
- ❑ Retains one-frame view, but inflexible vs Answer 1.
- ❑ Ccafe-GUI gets cluttered when:
 - Coder of X forgets to set visibility attribute on members of [Xsubs] component properties map.
- ❑ No specification extensions required, except to standardize interpretation of visibility property.

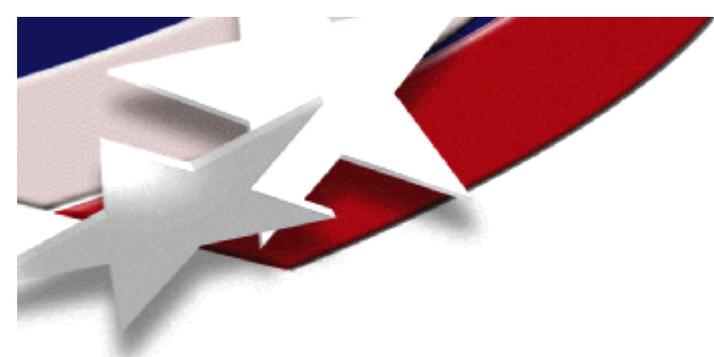




A3: Subassembly inside a component

- ❑ Must be orchestrated with help from non-CCA code which started the outermost AbstractFramework instance (frame) and supplies a clone frame to the container component, X_c . (See 2004 PPHEC paper).
- ❑ User needs to wire up only X_c . No [Xsubs] accessible.
- ❑ Hierarchical view, inner frames invisible.
- ❑ Ccafe-gui would be uncluttered (has no clue about the inner frame).
- ❑ Requires simple specification extension to eliminate non-CCA orchestration.

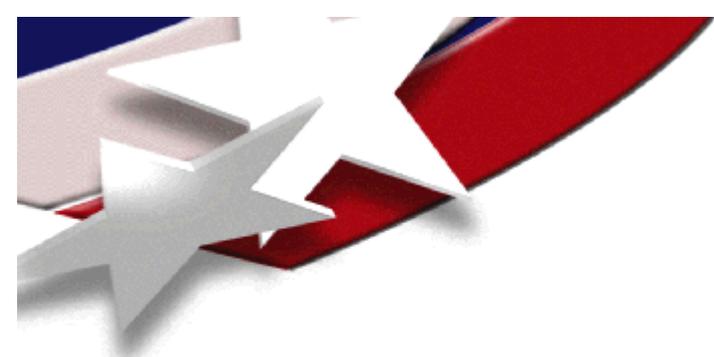




What's possible 1

- ❑ Tutorial could be expanded to include examples of approach 2.
- ❑ Tools could be created to automate generation of components that do approach 2 from ccafe-gui scripts.





What's possible 2

- ❑ Specification could be extended to include visibility and other rendering properties for assemblies. No big changes to frameworks are required. **Big GUI changes.**
- ❑ Component could advertise that it wants to be a container by **providing** a new standard port. (exactly one inner-frame handle per component). I.e. a frame setter-port.
- ❑ Component could advertise that it wants to be a container by **using** a new standard port. (many or dynamic sub-frames per component). I.e. a frame getter-port.
- ❑ Could extend Services to provide an inner-frame handle registration protocol.

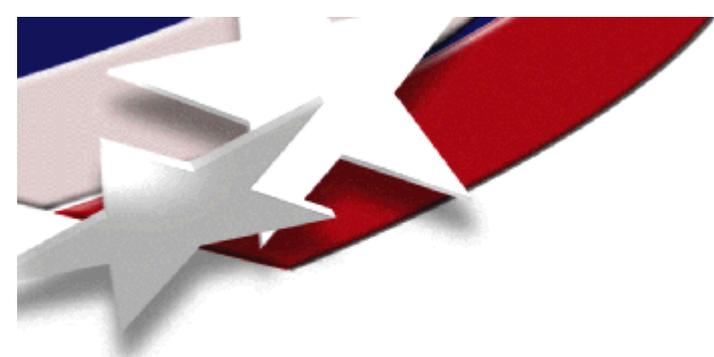




What we need to do

- Prioritize needs.
 - Is it better documentation, new specification details, or a rather different GUI?
 - ❖ Hiding means what?
 - ❖ Some GUI work is probably inescapable.
- Prioritize approaches.
 - If A3 first, specify the style of a component getting its inner-frame handles (Dei Causa, Eligete Quidquam!)





What we need to do (cont)

- ❑ Use CCARB to establish a working group on this topic chaired by someone who needs this capability and is willing to create support tools for it.
- ❑ Find funding– this is outside the specific deliverables for TASCs.

