

U153UK3FA : <@U5VD8S0HZ> when doing that pattern the ChildComponent module shouldn't need to know about the ParentComponent modules Msg type, the translation between the ChildComponent.Msg and it's meaning in the ParentComponent module's update function should be done there.

U153UK3FA : I.e. Define a ChildComponent.Msg constructor for the action you want to happen and translate it to it's ParentComponent.Msg equivalent in your update function

U153UK3FA : It's the price you pay for the encapsulation you want

U153UK3FA : If you don't actually need that encapsulation then you can save yourself a lot of pain by just having a single Msg type that you use everywhere

U153UK3FA : If you're new to Elm then definitely try having just a single Msg type and don't follow the 'components' pattern

U153UK3FA : New devs tend to over use the pattern and get themselves in to a mess

U5VD8S0HZ : <@U153UK3FA> the specific case I am working on is creating a general-use modal dialog component that I can pass content to. The way I am thinking about this, messages such as CloseDialog would be declared as part of the Dialog component and translated to the ParentComponent.Msg equivalent as you mentioned.

The problem is that messages that belong to the content (such as SelectCheckbox-type messages) belong to the parent component and remain unknown to the dialog component. Because of this, I haven't figured out a way to pass the content to the dialog component since the dialog component won't know about the content's Msg type.

U153UK3FA : Ah, you need to wrap the content Msgs in a Dialog Msg and have the caller unwrap them

U5VD8S0HZ : I am trying to parse what you just said. I'm not sure I completely get that.

U5VD8S0HZ : So, for example, right now in my content view, which exists in the parent component, I may have something like:``

```
li [ onClick SelectCheckbox checkbox.id ] []
```

where `SelectCheckbox : String` is a parent component Msg type

U5VD8S0HZ : Are you saying that I do something like:``

```
li [ onClick (DialogMsg SelectCheckbox checkbox.id) ] []
```

where DialogMsg is message type that is declared in the child component?

(sorry, my syntax is probably off here... not using an editor)

U5VD8S0HZ : or can you point me to an example of what you mean?

U3YDBDP4Z : Are there any good sources on good unit testing setups or strategies?

U5QJW0DDE : This is not correct. DCE is part of the advanced optimizations in Closure which Elm does not support. But you can use the simple optimizations of Closure, which don't do DCE.

U5QJW0DDE : Check out the Dev mailing list from yesterday if you want to see some of the issues regarding supporting the advanced optimization mode.

U0J8D9M2P : I'm implementing functions which uses latitude/longitude. The value range for latitude is from 0 to +/- 90. Is there a way to limit value e.g. with union type so that when I have incorrect value compiler will tell that it's not allowed?

U4872964V : No, there are no bounded number types in Elm. However, you can make your own module and wrap the number in a type and just expose the operations you need, and check the value inside the operations. It depends on what kind of usage of those coordinates you have

U0J8D9M2P : I have``

```
getPosition : Date -> Float -> Float -> { azimuth : Float, altitude : Float }
getPosition date latitude longitude = ...
```

U0J8D9M2P : So the best will be to check latitude/longitude values in function implementation and return Result, either with Ok or with Error.

U0J8D9M2P : Is that correct?

U4872964V : Yes, if `getPosition` can fail, I would return a `Maybe` or `Result`

U4872964V : so that the caller has to handle those cases

U2D7NUGS1 : How can I handle Error result of `Json.Decode`? I have a snippet like this: ```, on "opened-changed"

```
(Json.Decode.string
  |> <http://Json.Decode.at|Json.Decode.at> [ "detail", "value" ]
  |> Json.Decode.map StringHappened
)
```

...

where `StringHappened` is a constructor for my `Msg`. It just so happens that the `detail.value` is a `bool`. Currently in my app nothing happens. I want another variant of `Msg` to be used, say `ErrorHappened`. How can I do it?

U1CE9DL9H : so detail.value can possibly have two types?

U41NK9BM4 : Or maybe he wants to handle the bad value more gracefully?

U2D7NUGS1 : <@U41NK9BM4> yes.

U41NK9BM4 : But does it crash if arrives a bool type as `detail.value` ?