

Code Development of PTM for Adding Particle Behaviors

Xiaochun “Shiao” Wang
DWR Modeling Support Branch

Brief Review of PTM

Neutral buoyant particle:

- Move passively with flow, no active swimming velocity
- Make route decision based on flow split
- Always stay alive

Code Design in PTM

- Basic Classes in PTM:
 - WaterBody
 - Node
 - Particle

Code Design in PTM

- WaterBody (Channel, etc.)
 - Carry up/down node IDs
 - Store channel geometry info
 - Store channel hydrodynamics info
 - Calculate velocities

Code Design in PTM

- Node
 - Store connecting water bodies info
 - Store/calculate in/out flows
 - Roll dice for particles -- generate random numbers

Code Design in PTM

- Particle
 - Know node passed and water body currently in
 - Make decision where to go at junctions
 - Move itself—calculate X, Y, Z positions

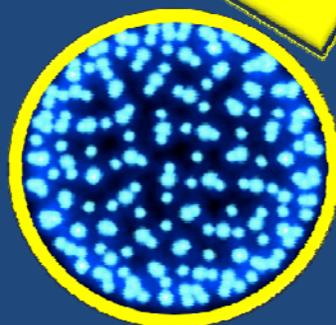
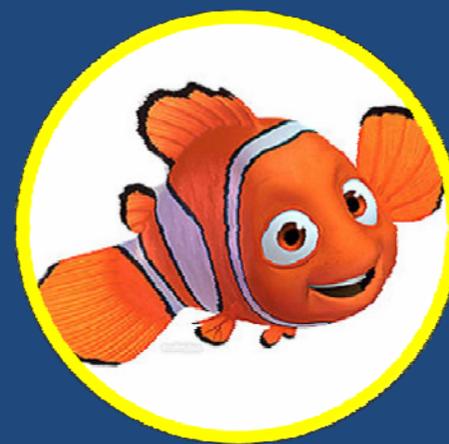
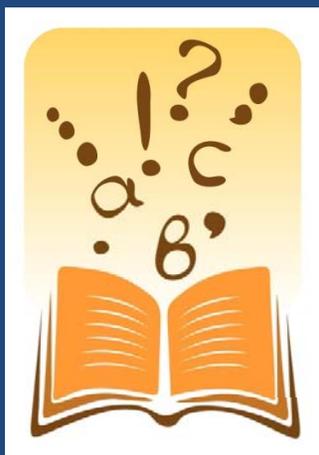
Adding Behaviors to Particle

Fish Behaviors Depend on Its environment, Including:

- Flows: magnitudes, directions, split ratios...
- Water quality: temperature, salinity, turbidity...
- Geometry: channel length, width...
- Water projects: operations of gate, barrier, PP.....
- Weather conditions: light intensity, wind, air temperature...
- Predators
-

Dependencies vary in both time and space!

Behavior Book



Helper

Particle

Particle Helpers

- Behavior relationships implemented in helpers
- Implementation independent from particles
- Particles movement observed by helpers
- When helps requested, helps provided

Helper Classes

- Route Decision Helper
 - Route Selection Rules—where to go
- Swim Helper
 - Swimming Rules—how to swim
- Survival Helper
 - Survival Rules—whether survive or not

Helper Hold Behaviors

- Basic behaviors
 - universal
- Special behaviors
 - under certain specific conditions
- All behaviors stored in HashMap with proper search key

Functionality of Helper Class

- Initialize, add, remove behaviors
- Lookup behaviors
 - Search behavior book to find instructions
- Help Particle
 - Handle help requests
 - Set behaviors for particle

Behavior Class: Implement Behavior Instructions

- One behavior per class
- Independent or inherited
- Same interface implemented for the group

Implement in PTM

- Scalability
 - Add behaviors, helpers, species
- Type safety
 - Avoid wrong data type or adding unintended behaviors
- Reusability
 - Maintain one set of code for doing the same things

Instantiate Helpers Object

```
public static void main(String[] args) {  
    .....  
    Helpers helpers = new Helpers(particleType)  
    .....  
}
```

Instantiate Specific Helper Objects

```
public class Helpers {  
    public Helpers(String particleType) {  
        switch (particleType){  
            case "SALMON":  
                routeHelper = new SalmonRouteHelper();  
                routeHelper.setBasicBehavior(new SalmonBasicRouteBehavior());  
                swimHelper = new SalmonSwimHelper();  
                .....  
                survivalHelper = new SalmonSurvivalHelper();  
                .....  
                break;  
            case "SMELT":  
                .....  
        }  
    }  
}
```

Inside Helper

```
public abstract class Helper<Key, Behavior> {  
    .....  
    public Behavior lookUp(Key key){...}  
    public setBasicBehavior(Behavior basic){...}  
    public addSpecialBehavior(Behavior special){...}  
    public removeSpecialBehavior(Key key){...}  
    .....  
    public abstract Key getKey(Particle p);  
    public abstract void help(Particle p);  
    .....  
}
```

Extend Helper

- `public class SalmonRouteHelper extends Helper<Integer, SalmonRouteBehavior> {...}`
- `public class SalmonSwimHelper extends Helper<Integer, SalmonSwimBehavior> {...}`
- `public class SalmonSurvivalHelper extends Helper<Integer, SalmonSurvivalBehavior> {...}`

Install Helper Objects to Particles

```
public static void main(String[] args) {  
    .....  
    Helpers helpers = new Helpers(Particle Type)  
    .....  
    for(int i=0; i<numberOfParticles; i++) {  
        if (helpers != null)  
            helpers.setHelpersForParticle(particleArray[i]);  
    }  
    .....  
}
```

Install Helper Objects to Particles

```
public class Helpers {  
    .....  
    public void setHelpersForParticle(Particle p){  
        p.installRouteHelper(routeHelper);  
        p.installSwimHelper(swimHelper);  
        p.installSurvivalHelper(survivalHelper);  
    }  
    .....  
}
```

Call for Help

In Particle Object:

- If route decision help needed:
routeHelper.help(this)
- If swimming help needed
swimHelper.help(this)
- If survival help needed
survivalHelper.help(this)

Provide Help

```
public class SalmonRouteHelper extends
    Helper<Integer, SalmonRouteBehavior> {
    .....
    public void help(Particle p){
        super.getBehavior(p).makeRouteDecision(p);
    }
    .....
}
```

Add Special Behaviors

- Create a special behavior
 SalmonGSJRouteBehavior
- Add the behavior to helper:

```
public static void main(String[] args) {  
    .....  
    Helpers helpers = new Helpers(Particle Type)  
    helpers.addRouteBehavior(NodeID,  
                             new SalmonGSJRouteBehavior());  
    .....  
}
```

Summary

- Delegate behavior calculations from Particle to Helper
- Helper as observer installed in particle
- Three types of helpers to help particle to behave
- Specific behaviors can be added to model w/o touch existing particle code
- Scalable for other fish species, helpers and behaviors

Future Work

- Route decision models
- Survival models
- Swimming behaviors—NMFS
- Special swimming behaviors related to turbidity for smelt--RMA