

## (2) Yuho: DSL for Criminal Law (Credits to Gabriel Ong, our Tech Dev Director)

Yuho provides the following three products:

1. Yuho, a **DSL** made to be readable and codeable by students and lawyers
2. **Formalised semantic** for legal reasoning modelled after the syntactical patterns of the law
3. **Transpiler** that transpiles to the following targets, and for the following purposes

JavaScript	Browser simulations and extensions
JSON	REST APIs
R	Data modelling and visualisation
Alloy	Formal verification
Whiley	Decision logic
F*	Proof backends

## (5) DSL for Criminal Law: Yuho

- Yuho (*derived from yume ho 夢法 which means ideal law*)
- Many attempts have been made to programmatically represent the law, for many purposes
  - Law-related: Catala, CCLAW Natural L4
  - Business-related: Blawx, Morphir
- What would such decision logic look like for SG Criminal Law?
- Define a flexible syntax that transpiles to multiple target outputs, embedding logic processing and exception handling into its construction by default
- Yuho ships with the following
  - Yuho, a DSL made to be readable and codeable by lawyers (TESTED ON 7 LLB students and still going!)
  - Formalised semantic for legal reasoning modelled after the syntactical patterns of the law (Alloy, Whiley, F\* thus far, more to come)
  - Transpiler that transpiles to the following targets
    - JavaScript: browser simulations and extensions
    - JSON: REST APIs
    - R: data modelling and visualisation
    - Alloy: formal verification
    - Whiley: decision logic
    - F\*: proof backends
- For more details can see documentation here (<https://github.com/gongahkia/yuho>)

```
scope sampleCheating {
```

```
// STRUCT DEFINITIONS
```

```
struct statute {  
    int sectionNumber,  
    string sectionDescription,  
    string definition,  
    punishmentForCheating result,  
}
```

```
struct punishmentForCheating {  
    punishment ofGeneric,  
    punishment ofPersonation,  
}
```

```
punishment {  
    duration imprisonmentDuration,  
    pass | money fine,  
    pass | string supplementaryPunishment,  
}
```

```
// STRUCT LITERALS
```

```
statute aStatuteOnCheating := {  
    sectionNumber := 415,  
    sectionDescription := "Cheating",  
    definition := "Whoever, by deceiving any person, fraudulently or dishonestly induces the person so deceived to deliver any property to any person, or to consent that any person shall  
    result := {  
        ofGeneric := {  
            imprisonmentDuration := 3 year,  
            fine := pass,  
            supplementaryPunishment := pass,  
        },  
        ofPersonation := {  
            imprisonmentDuration := 7 year,  
            fine := pass,  
            supplementaryPunishment := "A person convicted of cheating by personation shall be disqualified for such period as the court may order from holding any office of profit under  
    },  
}
```

```
const aStatuteOnTheft = {
  sectionNumber: 378,
  sectionDescription: "Theft",
  definition: "Whoever, intending to take dishonestly any movable property out of the possession of any person without that person's consent, moves that property in",
  result: {
    ofGeneric: {
      imprisonmentDuration: "3 year",
      fine: null,
      supplementaryPunishment: null
    },
    ofMotorVehicle: {
      imprisonmentDuration: "7 year",
      fine: null,
      supplementaryPunishment: "A person convicted of an offence under this section shall, unless the court for special reasons thinks fit to order otherwise, be",
    },
    ofDwellingHouse: {
      imprisonmentDuration: "7 year",
      fine: null,
      supplementaryPunishment: null
    },
    ofClerkOrServant: {
      imprisonmentDuration: "7 year",
      fine: null,
      supplementaryPunishment: null
    },
    afterPreparationCausingDeath: {
      imprisonmentDuration: "10 year",
      fine: null,
      supplementaryPunishment: "caning with not less than 3 strokes"
    }
  }
};
```

```
sig Duration {}
sig Money {}

sig Punishment {
  imprisonmentDuration: lone Duration,
  fine: lone Money, // Using lone to allow for an optional field
  supplementaryPunishment: lone String // Using lone to allow for an optional field
}

sig PunishmentForTheft {
  ofGeneric: lone Punishment,
  ofMotorVehicle: lone Punishment,
  ofDwellingHouse: lone Punishment,
  ofClerkOrServant: lone Punishment,
  afterPreparationCausingDeath: lone Punishment
}

sig Statute {
  sectionNumber: Int,
  sectionDescription: String,
  definition: String,
  result: lone PunishmentForTheft
}

fact aStatuteOnTheft {
  let p1 = Punishment {
    imprisonmentDuration = Duration,
    fine = none,
    supplementaryPunishment = none
  },
  p2 = Punishment {
    imprisonmentDuration = Duration,
```

```
1  type duration = string
2  type money = string
3
4  type punishment = {
5      imprisonmentDuration : duration;
6      fine : unit | money;
7      supplementaryPunishment : unit | string;
8  }
9
10 type punishmentForTheft = {
11     ofGeneric : punishment;
12     ofMotorVehicle : punishment;
13     ofDwellingHouse : punishment;
14     ofClerkOrServant : punishment;
15     afterPreparationCausingDeath : punishment;
16 }
17
18 type statute = {
19     sectionNumber : int;
20     sectionDescription : string;
21     definition : string;
22     result : punishmentForTheft;
23 }
24
25 let aStatuteOnTheft : statute = {
26     sectionNumber = 378;
27     sectionDescription = "Theft";
28     definition = "Whoever, intending to take dishonestly any movable property out of the possession of any person without that person's consent, moves that property";
29     result = {
30         ofGeneric = {
31             imprisonmentDuration = "3 year";
32             fine = ();
33             supplementaryPunishment = ();
34         };
35         ofMotorVehicle = {
36             imprisonmentDuration = "7 year";
37             fine = ();
```

```
3 aStatuteOnTheft <- list(  
4   sectionNumber = 378,  
5   sectionDescription = "Theft",  
6   definition = "Whoever, intending to take dishonestly any movable property out of the possession of any person without that person  
7   result = list(  
8     ofGeneric = list(  
9       imprisonmentDuration = 3,  
10      fine = NULL,  
11      supplementaryPunishment = NULL  
12    ),  
13    ofMotorVehicle = list(  
14      imprisonmentDuration = 7,  
15      fine = NULL,  
16      supplementaryPunishment = "A person convicted of an offence under this section shall, unless the court for special reasons th  
17    ),  
18    ofDwellingHouse = list(  
19      imprisonmentDuration = 7,  
20      fine = NULL,  
21      supplementaryPunishment = NULL  
22    ),  
23    ofClerkOrServant = list(  
24      imprisonmentDuration = 7,  
25      fine = NULL,  
26      supplementaryPunishment = NULL  
27    ),  
28    afterPreparationCausingDeath = list(  
29      imprisonmentDuration = 10,  
30      fine = NULL,  
31      supplementaryPunishment = "caning with not less than 3 strokes"  
32  )  
33 )  
34 )  
35
```

# Looking into the Future...

- Increase the number of targets for transpilation and use cases
- Beyond criminal law
- Continue with further testing on more students and use more rigorous testing methods  
(Currently, Yuho has been tested on 7 LLB students)



## (2) Yuho: DSL for Criminal Law (Credits to Gabriel Ong, our Tech Dev Director)

For more details can see documentation here (<https://github.com/gongahkia/yuho>)

