

An Architecture of Digital Gastronomy Software Design Document

Date: 16/03/2020

Name: Noam Domovich

ID: 204083893.

Coordinator: Shimon Bezalel.

Lab: Hybrid Lab School of Engineering and Computer Science at the Hebrew University, Prof.
Amit Zuran .

Table of content

1. INTRODUCTION	4
1.1 Purpose	4
1.2 Scope	4
2. SYSTEM OVERVIEW	4
3. SYSTEM ARCHITECTURE	5
3.1 Architectural Design	5
Design Pattern	5
Tech Stack	5
3.2 Decomposition Description	5
Dish Control/ Reporting Through Parameterization	5
Meta-Recipe Design	5
3rd Party APIs	6
Cookwares APIs	6
4. DATA DESIGN	6
4.1 Data Description	6
4.2 Data Dictionary	6
Meta-recipe json schema	6
Aroma json schema/shcema	8
Taste json schema/Eample	8
Cookware json schema/Example	9
5. COMPONENT DESIGN	10
Controllable Spider Chart	10
Aroma, Taste	10
Dish Report	10
Environmental Footprint	10
Dish Renderer	11
Dish Controllers	11
Slider	11
Distribution Control	12
Free Hand	12
Control Point	12
Sliders / buttons	12
Menu	12
6. HUMAN INTERFACE DESIGN	13

6.1 Overview of User Interface	13
cook perspective:	13
7. APPENDICES	14

1. INTRODUCTION

1.1 Purpose

The project purpose is developing a research tool for Digital Gastronomy (DG), by creating the outlines of a system. the product will cover APIs with other databases and architectures to be used for analyzing the properties of digital gastronomy, and how to improve the quality of this process.

1.2 Scope

Name: An Architecture of Digital Gastronomy

Focus:

- Research about the interactions and protocols for the needs of DG
- Collections and analysis for datasets such as aroma and taste intensities, definition of meta-recipe, etc. in digital context, and create architectures to be used for cooks while they cook.
- Tools for the cook in real-time environments of dish manufacturing.

2. SYSTEM OVERVIEW

A digital gastronomy intended to be for help to cook and give him a tool and option that he never had Before. Therefore a digital gastronomy process, in any phase we will include cook and the digital agent at the same time. The system will lead the cook during the cooking and give him cooking tools to cook the dish.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

Design Pattern

MVC : modal view Controller

Modal

Will take care about all application data: Aroma and taste intensity , recipes, ingredients status in the inventory , kitchen's cookware

View:

Will take care about the communication with between the cook and the application

Controller:

Will take care about the cook requests and preferences for the dish cooking process

Tech Stack

- Node.js - for back end
- Maybe ionic for front end

3.2 Decomposition Description

Dish Control/ Reporting Through Parameterization

This unit will collect and analyze all the dish data:

- 1) It will follow after the recipe instructions
- 2) Ask for the requested ingredients
- 3) Send the operating instructions for the cookwares
- 4) Adjust the cook's preferences for the dish.

Meta-Recipe Design

This unit will hold a template for recipe. According to this template the 'Dish control' will send command and requests to the inventory, cookware and will lead the cook during the cooking.

3rd Party APIs

This unit will demonstrate the aroma and taste intensity for each ingredient. The intensity is based on data which is analyzed and stored in the database.

Cookwares APIs

This unit will hold the ways to communicate with the cookware. for example the robotic arms. What is the pre-time it could for turning on? What time it takes to print it's part in the dish, what the required locations for the dish and ingredient , and so on.

4. DATA DESIGN

4.1 Data Description

The main DB's data object:

- Inventory status - what ingredient exists in the inventory
- Recipes - set of instructions and requirements for the cooking process
- Ingredient aroma and taste - analysis of aroma and taste for each ingredient
- Cookware - what cookware the cook can use

4.2 Data Dictionary

Meta-recipe json schema

example:

```
{
  "recipes": [
    {
      "id": 1,
      "name": "omelette",
      "ingredients": [340, 222, 500] ,
      "tools": [1,2,3,4,5,6,7,8,9,10],
      "instructions": :
      [
        {
          "step": 1 ,
```

```

    "during" : {
      "max_time": "80 min",
      "min_time" : "5 min"
    },
    "range" : {
      "most_before": "500 min",
      "most_recently" : "5 min"
    },
    "action": "Season the beaten eggs well with salt and pepper.",
    "step_tool": [3,2] ,
    "step_ingredient" : [340]
  }
]
}

```

For schema take a look at appendix

Aroma json schema/shcema

example:

```
{
  "ingredients": [
    {
      "category": "animalproduct",
      "entity_id": 0,
      "entity_alias_readable": "Egg",
      "statistical_aroma": {
        "Uncategorised": 0.16,    "Decayed": 0.3,    "Sweet": 0.1,    "Woody": 0.12,
        "Medicinal": 0.31,    "Sulfidic": 0.23,    "Fruity": 0.05,    "Smoky": 0.19,
        "Floral": 0.08,    "Citrus": 0.07,    "Mint": 0.04
      },
      "culinary_1_aroma": {
        "Uncategorised": 0.15,    "Maillard": 0.16,    "Terpene": 0.12,    "Veg": 0.17,
        "Fruity": 0.07,    "Dairy": 0.19,    "Floral": 0.08,    "Sulphur": 0.1,    "Marine":
        0.28,    "Sour": 0.21,    "Phenol": 0.15,    "Pungent": 0.13,    "Savory": 0
      },
      "culinary_2_aroma": {
        "Uncategorised": 0.15,    "Meaty": 0.24,    " Nutty": 0.17,    "Medicinal": 0.14,
        "Chocolate": 0.22,    "Wood": 0.05,    "Spice": 0.02,    "Herbaceous": 0.03,    "
        Berry":0.06,    "Dairy": 0.19,    "Green": 0.12,    "Roasted": 0.1,    "Petrol": 0.6,
        "Floral": 0.08,    "Caramel": 0.07,    "Sulphur": 0.1,    " Tree fruit": 0.01,
        "Tropical":0.04,    "Marine": 0.28,    "Earthy": 0.29,    "Fruit like": 0.42,
        "Melon": 0.0,    "Sour": 0.21,    " Citrus": 0.12,    "Phenol": 0.15,    "Pungent":
        0.13,    "Toasted": 0.0,    "Smoke": 0.0,    "Savory": 0
      }
    }
  ]
}
```

Taste json schema/Eample

exmaple:

```
{
  "ingredients": [
    {
      "category": "animalproduct",
      "entity_id": 0,
```



```
"entity_alias_readable": "Egg",
"statistical_taste": {
  "Bitter": ??
  "Salty": ??
  "Sour" : ??
  "sweet " : ??
  "Umami" : ??
},
```

Cookware json schema/Example

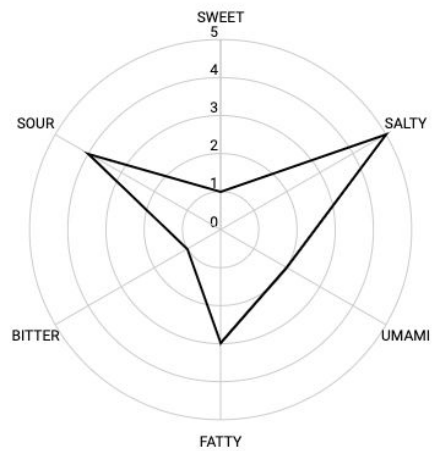
```
{
  "cookware": [
    {
      "id": 1,
      "name": "robotic arm",
      "pre-procssing" : [
        {
          "action" : "turn on"

        },
        {
          "action": "Filling containers"
        }
      ],
      "limits" : [
        {}
      ]
    }
  ]
}
```

5. COMPONENT DESIGN

5.1. Controllable Spider Chart

a. Aroma, Taste

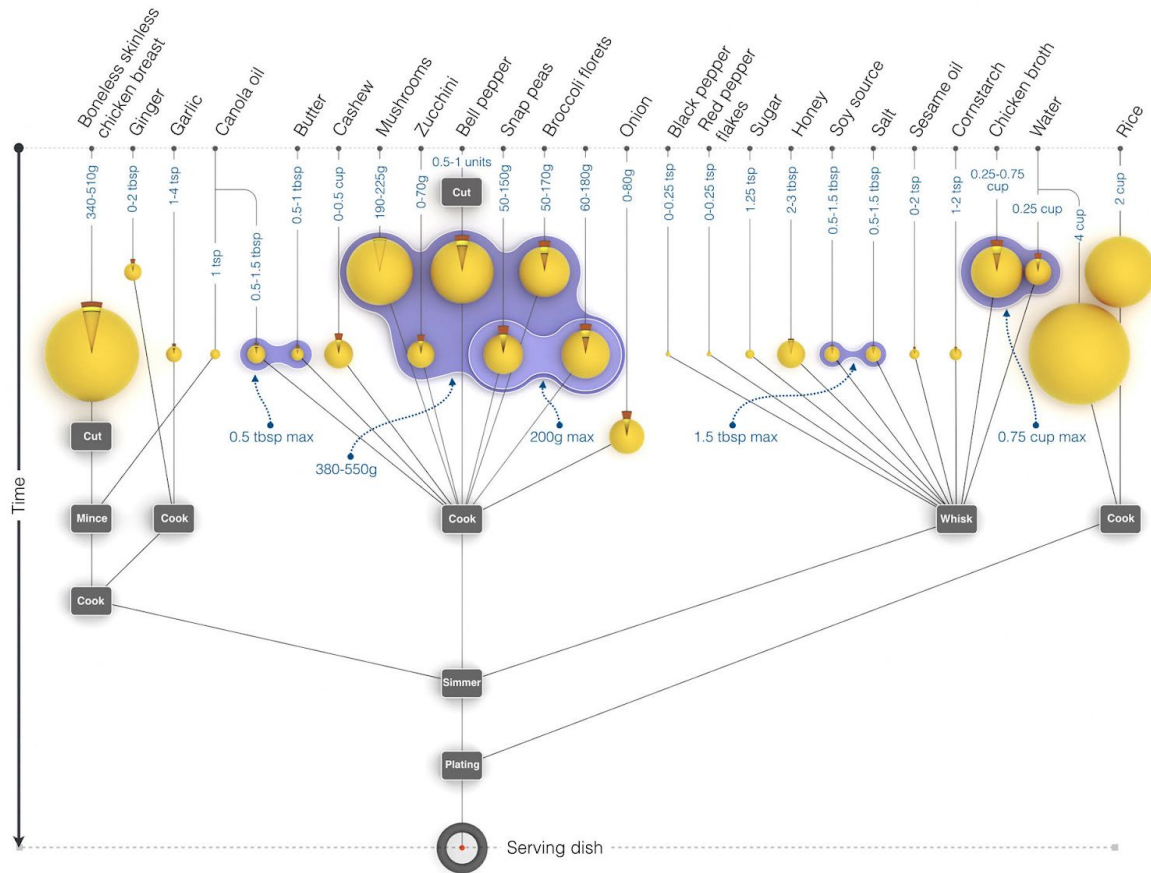


5.2. Dish Report

a. Environmental Footprint

A component that show for the user the dish progress and allow him to “edit” parts from the dish (for example, geometric change by user touch)

5.3. Dish Renderer

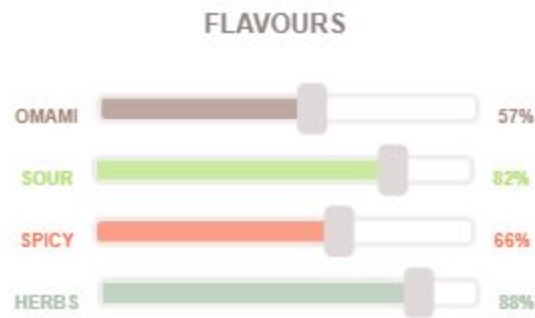


A component that analyzes any dish for small units. So the system could work with the dish units.

5.4. Dish Controllers

a. Slider

The user can define the exact flavours , spices and ect. For the dish by sliders.



b. Distribution Control

- i. Mechanism for spices (and relevant ingredients) distribution in on the dish

c. Free Hand

- i. A mechanism for the cook allows him to add and change freely details in the dishy

5.5. Control Point

Sliders / buttons

for fine tuning and geometrical compute - as we showed below

Menu

dish selection for cook/costumer , with all the restaurants options

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

cook perspective:

Shtift setup

- Dish selection
- Inventory assessment
 - Update aroma/taste
 - Confirm subtitidue
- Confirm ingredient

Mise en place

- (Manual preprocessing) update data to app
- Digital tool setup
- Day storage / station setup

Dish manufacturing

- Dish order
- Restriction filter
- Taste progression / customization
- Form finalization
- Dish illustration

7. APPENDICES

Meta -recipe Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "recipes": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
            "id": {
              "type": "integer"
            },
            "name": {
              "type": "string"
            },
            "ingredients": {
              "type": "array",
              "items": {}
            },
            "tools": {
              "type": "array",
              "items": {}
            },
            "instructions": {
              "type": "array",
              "items": [
                {
                  "type": "object",
                  "properties": {
                    "step": {
                      "type": "integer"
                    },
                    "during": {
                      "type": "object",
                      "properties": {
                        "max_time": {
                          "type": "string"
                        },
                        "min_time": {
```

```

        "type": "string"
    },
    },
    "required": [
        "max_time",
        "min_time"
    ],
    },
    "range": {
        "type": "object",
        "properties": {
            "most_before": {
                "type": "string"
            },
            "most_recently": {
                "type": "string"
            }
        },
    },
    "required": [
        "most_before",
        "most_recently"
    ],
    },
    "action": {
        "type": "string"
    },
    "step_tool": {
        "type": "array",
        "items": {}
    },
    "step_ingredient": {
        "type": "array",
        "items": {}
    },
    },
    "required": [
        "step",
        "during",
        "range",
        "action",
        "step_tool",
        "step_ingredient"
    ],
    },
    }
}

```

```
    }  
  },  
  "required": [  
    "id",  
    "name",  
    "ingredients",  
    "tools",  
    "instructions"  
  ],  
}  
]  
}  
}  
}  
  "required": [  
    "recipes"  
  ],  
}  
}
```