Example Project: Microwave Oven Project

Illustrates the application of the steps of the Software Development Process (SDP)

TABLE OF CONTENTS

Section **Page**

1. CONCEPT DOCUMENT

1.1 INTRODUCTION

1.1.1 Description

The Pulsar Microwave Oven is a convenient kitchen appliance that allows users to cook, heat, or defrost food items in a relatively short amount of time. The Pulsar offers users a choice of several preset cooking programs for common food items. The user may also set cooking times manually. In addition, a digital clock is provided as well as a timer that will sound a buzzer after a previously entered length of time has passed (without the oven heating).

1.1.2 Purpose

The Pulsar Microwave Oven offers users a quick, convenient alternative to standard conventional ovens by allowing users to perform many common cooking tasks in a shorter amount of time.

1.1.3 Scope

The Pulsar is ideal for any home or office setting. It provides a quick and easy way to for anyone to cook, heat up, or defrost a variety of food and beverage items.

1.1.4 Goals/Objectives

Below is a list of the goals and objectives for this project:

- Provide a user-friendly alternative to conventional ovens.
- Provide a simple interface that offers an array of features while maintaining maximum usability.

1.1.5 Users

The Pulsar Microwave Oven is ideal for household kitchen environments where individuals would prefer a more convenient means of performing basic cooking tasks without the complication and time required by a conventional oven. The unit will be robust yet easy-to-use. All but the youngest members of a household will be able to use it without difficulty.

The Pulsar would also be an appropriate choice for offices or other work environments where conventional ovens would be impractical for reasons of convenience, safety, and space availability.

1.1.6 Capabilities

The Pulsar Microwave Oven provides the following:

- Manual cooking time options
- One-touch cooking presets
- LCD screen
- Sound alerts
- Heating compartment lighting
- Rotating plate
- Digital Clock
- Timer

1.1.7 Planned Future Extensions

Planned extensions to this project are listed below:

- The ability to detect metal food containers and utensils
- Voice alerts
- Voice recognition/activation
- Reduced weight and size

1.2 NON-FUNCTIONAL CHARACTERISTICS

1.2.1 Performance

The Pulsar performs a cooking task chosen by the user from one of the offered presets or cooks for an amount of time desired and set by the user.

1.2.2 Interface

The user interface is so simple that an individual with little or no electronic background will have no difficulty in its use. The interface offers the following features:

- Separate keys for each of the presets
- Separate keys for each power level
- Start, stop, and clear keys

- Keys to set and activate digital timer and clock
- Door sensor that shuts down the microwave if the cooking is in process when the door is opened

The interface also allows the user to manually set the timer and clock as well as program additional user-defined presets.

1.2.3 Reliability

The Pulsar only performs the functions input by the user. If the user chooses the wrong preset or inputs an incorrect cooking time, the Clear key can be used to terminate the incorrect function. The user can then reselect the desired function. Should an internal error occur, safety features included in the control software will terminate the current cooking process. An error message will be generated indicating that the user should contact the manufacturer for instructions.

1.2.4 Quality

The Pulsar is convenient, user friendly, and time saving. Its durable construction and ease-of-use will insure consistent and satisfactory results, as well as a lengthy usable life span. Both the hardware and software associated with this product will be developed using proven processes to guarantee that a high quality product will be produced.

1.2.5 Security

Both firewalls and antivirus software will be utilized during the development of the product to provide a secure software development environment. There is no classified material in the software under development.

1.2.6 Maintainability

Software:
The software will be developed using well-defined processes that ensure good programming practices are followed. This will make the software easy to modify for future enhancements.

Hardware:
1) Use microwave as directed in the user handbook.
2) Clean often.
3) Food and beverage items must be removed from metal containers or aluminum foil before heating.

1.3 WORKING ENVIRONMENT

1.3.1 Current

The Pulsar control software and internal components are currently running in the P110 model.

1.3.2 Potential

The control software and internal components will be expanded for use in the P111 and P112 as well as other future models.

1.4 OPERATIONAL SCENARIOS

Microwave Popcorn: Using Popcorn Preset Function

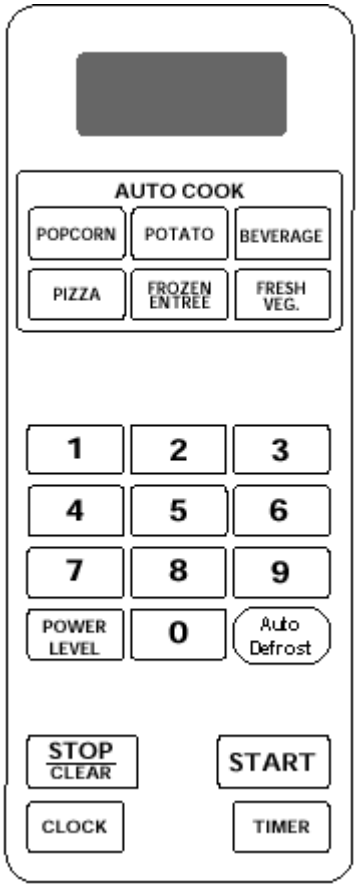
1) User opens the microwave door.
2) Popcorn is placed in the heating chamber.
3) User then presses the Popcorn preset key.
4) The timer is automatically set to 3 min. 30 sec.
5) The microwave cooks for the set amount of time.
6) Beeper sounds indicating cooking time has elapsed.
7) User opens the door and removes popcorn.

Microwave Popcorn: Manual Timer Setting

1) User opens the microwave door.
2) Popcorn is placed in the heating chamber.
3) User manually sets the cooking timer for 5 min.
4) User opens the door before cooking is finished (microwave stops automatically).
5) User checks for a burning odor.
6) User closes the door and presses start to resume cooking.
7) User opens door again before the time has elapsed.
8) User determines that popcorn is finished.
9) User removes popcorn and presses Clear to reset the timer.

1.5. PRELIMINARY USER'S MANUAL

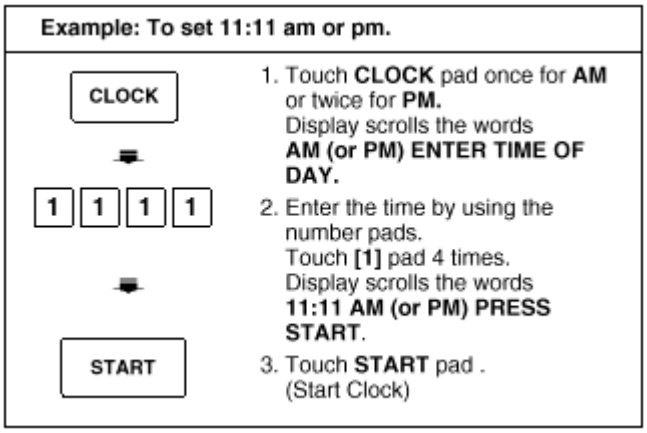
1.5.1 USER INTERFACE



1.5.2 OPERATIONAL INSTRUCTIONS

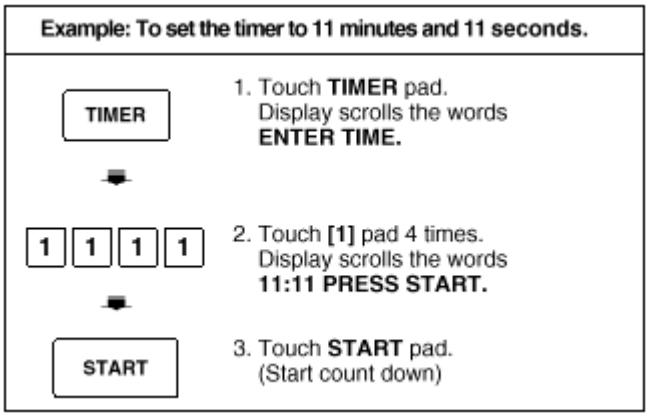
Setting the Clock

If you are using the microwave oven for the first time or the clock has been reset by a power surge or outage, the words ENTER TIME will appear in the display. Follow the procedure below to set the time.



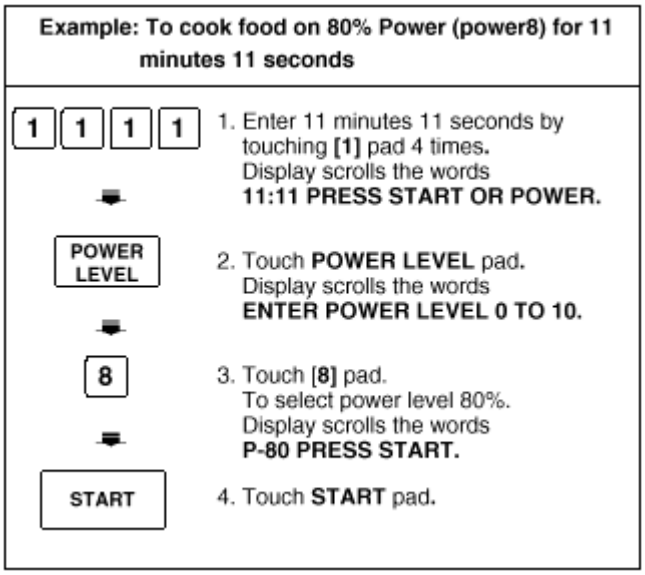
Timer

The oven's timer function allows the user to set a time that counts down without the oven heating. To set the oven's timer, follow the steps in the diagram below.



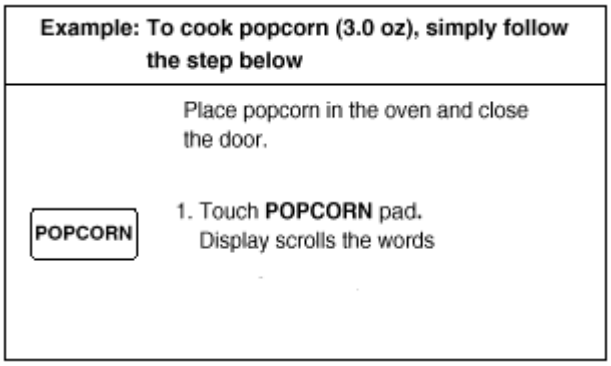
Timed Cooking

The timed cooking feature allows the user to cook for a desired amount of time at a desired power level. The power level is set by default to HIGH. To cook at another power level for a desired amount of time, follow the example given below.



Auto Presets

The microwave's interface provides several presets that allow a user to cook certain common food items automatically. To use these presets, follow the example for the Popcorn preset given below.



Auto Cook Guidelines

POPCORN

1) Follow package instructions.
2) Use only popcorn packages that are made especially for microwave cooking.
3) Heat only one package at a time.

POTATO

1) Place in center of the oven on paper towel.
2) Pierce potato several times with fork.
3) After cooking, let stand for 5 minutes.

BEVERAGE

1) Use mug or microwave-safe cup.
2) Stir after reheating.

PIZZA

1) Place the refrigerated pizza on the plate in the center of the microwave oven.

FROZEN ENTRÉE

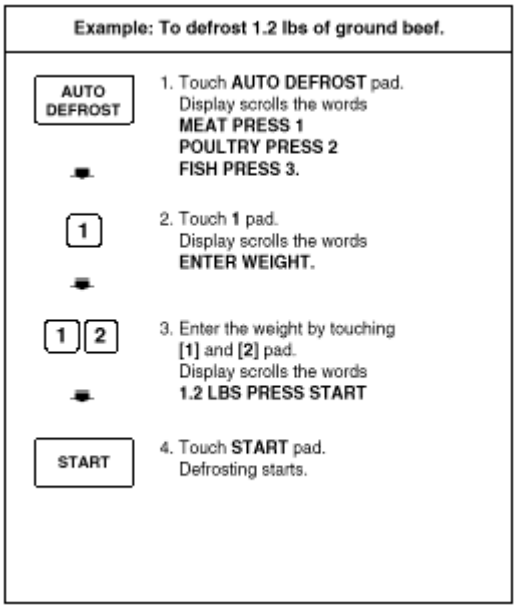
1) Follow the package instructions.
2) Example of one frozen entree is lasagna with meat sauce.

FRESH VEGETABLE

1) Prepare vegetables (wash, cut).
2) Add water as needed.

Auto Defrost

The microwave is programmed with three auto defrost settings: MEAT, POULTRY, FISH. The Auto Defrost feature automatically sets the defrost time for the given food item depending on the weight entered by the user. An example of the use of the Auto Defrost feature is given below:



1.5.3 CLEANING INSTRUCTIONS

INSIDE OF THE OVEN

1) Use a damp cloth to wipe out crumbs and spillovers.
2) It is important to keep the area between door and the front of the heating chamber clean to insure that the door closes properly.
3) Remove grease with a wet cloth, then and dry.
4) Do not use harsh detergent or abrasive cleaners.

GLASS TRAY

The glass tray can be washed by hand or in the dishwasher.

CONTROL PANEL

1) Wipe with a damp cloth followed immediately by a dry cloth.

OUTSIDE OVEN SURFACE

1) Clean with damp cloth.

1.5.4 TROUBLESHOOTING

If a problem occurs with the unit, use the guide below and try the solution for each problem that occurs.

OVEN WILL NOT START AT ALL

1) Make sure electrical cord for oven is plugged in.
2) Make sure the door is shut securely.
3) Check operation instructions to make sure current operation is being performed correctly.

UNEVEN COOKING OR POOR DEFROSTING

1) Make certain the Cooking time/Cooking power level is suitable for the item being heated.
2) Make sure food is turned or stirred properly.

OVERCOOKED FOODS

1) Make certain the Cooking time/Cooking power level is suitable for the item being heated.

UNDERCOOKED FOODS

1) Make certain food is defrosted completely.
2) Make certain the Cooking time/Cooking power level is suitable for the item being heated.

2. PROJECT PLAN

2.1 INTRODUCTION

The purpose of this document is to provide the up front planning necessary to effectively manage, develop, test, and document the software that will be used to control the Pulsar Microwave Oven (Model P110). The goal is to complete the project on schedule and within the allocated budget. The software organization has been given a budget of $350,000 and a six-month period of time to complete the software development including all testing and documentation. The budget is to be used buy all of the hardware and software needed to support the software development as well as to pay all direct labor costs associated with this project.

This plan identifies the members of the software development team and their respective responsibilities. Potential risks to this software development effort are listed along with a plan to minimize the impact of each risk should it occur. Hardware and software resources required to develop the software are also identified. A work breakdown is included as part of the plan along with the schedule that will be used to complete out this work. Note: Both the work breakdown and the schedule are fully defined in section 3. Project Schedule.

This project plan is considered to be a living document that will continue to be updated and modified as necessary during the life of this project.

2.2 SCOPE

This project plan only covers the software portion of the system. See Hardware Development Plan (D10003) for detailed planning information on the hardware portion of this project.

Note: The hardware and software development teams will need to work together throughout the development of this product to ensure that all project milestones are met.

2.3 REFERENCED DOCUMENTS

D10001	Software Quality Assurance Plan	Rev D	August 25, 2000
D10002	Software Configuration Management Plan	Rev C	October 10, 2000
D10003	Hardware Development Plan	Rev A	November 15, 2000
D10004	Staff Development Plan	Rev -	December 12, 2000
D10005	Verification and Validation Plan	Rev -	January 19, 2001
D10006	Maintenance Plan	Rev -	TBD

2.4 PROJECT ORGANIZATION

Below is a list of the personnel who will initially support the Pulsar Microwave Oven software project along with their respective positions. Following the list of personnel is a detailed description of the responsibilities defined for each of the identified roles. This list, along with the corresponding project scheduling information, will be updated as necessary to reflect any changes in project personnel during the software development life cycle.

Software Development Team

Dr. Sheila Williams	Software Manager
Andy Largent	Lead Software Engineer
Don David	Software Engineer
Jack Green	Software Engineer
Bob Sanders	Software Engineer

2.4.1 Software Manager Responsibilities

The Software Manager is primarily responsible for people, processes and tools and for providing these resources to the team as necessary to accomplish team tasks.

Team support:

a) Provide appropriately skilled personnel for team assignments.
b) Forecast and procure tools and laboratory resources.
c) Provide technical guidance/oversight to team members and monitor/support team members in task accomplishments.
d) Participate in weekly team meetings.
e) Participate in all technical reviews per defined processes.
f) Reassign functional personnel upon team task completion.
g) Foster a team environment and facilitate technical communication.

People:

a) Conduct or participate in employee salary reviews (with input from Lead Software Engineer).
b) Establish and communicate work performance expectations and track/review progress.
c) Prioritize tasks and coordinate with Lead Software Engineer whenever possible.
d) Address maintenance of adequate staffing levels. Monitor individual workloads to assure that tasks can be completed on schedule.
e) Provide the environment to improve knowledge, attitude, and skill. Support employee growth and development by providing appropriate coaching. Provide employee training in processes, tools, technology, team and people skills.
f) Resolve personnel problems or conflicts as appropriate.
g) Remove barriers, promote teamwork and empower people to improve performance.

Processes:

a) Provide and ensure the documentation and institutionalization of standards and processes. Ensure the proper use of reliable and common processes.
b) Work with other leaders to build greater consistency (common processes/procedures) among functional activities.
c) Estimate future conditions, opportunities and problems. Provide processes and training to handle these.
d) Implement continuous quality improvement on processes.
e) Assure quality, integrity and uniformity of processes used.
f) Sign off on all required documentation releases and problem reports focusing on quality and process.
g) Ensure that defined processes are followed, alternate design and cost trade studies are evaluated, maintainability, producability and testability, issues are properly addressed and product functionality requirements are satisfied.
h) Resolve tool and process related issues
i) Ensure timely and appropriate resolution of all identified problems.
j) Review and sign-off on Employee time charging.

Tools:

a) Determine and provide the appropriate tools to implement processes within the budget constraints.

2.4.2 Lead Software Engineer Responsibilities

The primary responsibility of the Lead Software Engineer is to lead and oversee their team in the organizing, planning, and plan execution activities as they execute the responsibilities as stipulated by the Software Manager. The Lead Software Engineer's responsibilities include but are not limited to the following:

a) Make team member assignments. Everyone should always know exactly what is expected of them and when it is expected. Also, everyone should know what is to be worked upon completion of their current task.
b) Help team members understand how their individual roles and assignments fit into the bigger picture.
c) Monitor team performance against technical, cost and schedule goals and help the team establish recovery/action plans as required to correct deviations from the team plan.
d) Anticipate problems and lead and implement planning to prevent their occurrence and/or to minimize their impact.
e) Develop schedules and report schedule status in weekly activity reports and team meetings. Have adequate metrics to know and communicate plans versus actuals for the team to meet the team's commitments. Keep the Software Manager informed.
f) Organize and conduct team meetings, provide guidance and leadership to the team to assist them in meeting their commitments.
g) Lead the team to ensure decisions are made in timely manner and problems are brought to the Software Manager's attention that are out of the scope of the team.
h) Act as focal point, facilitator and arbitrator for technical issues, work around planning, technical redirection, requirements review and redirection,

implementation issues and quality improvement activities, as appropriate, to meet the requirements and commitments for the team.

i) Ensure technical quality and on-time delivery of all team products. Ensure alternate design and cost trade studies are evaluated, maintainability, producability and testability issues are properly addressed and product functionality requirements are satisfied.
j) Represent the team with regard to production and changes to the schedule and/or baseline. Work with the team to gain consensus for changes to activities that impact them.
k) Ensure that standards, design procedures, and project philosophy are accessible, understood, and followed.
l) Ensure proper training of team members (overview, OJT, and formal training as necessary).
m) Provide input to the Software Manager for use in the performance evaluation of the team members in the normal employee review cycle.
n) Apply initiative and leadership, creativity and enthusiasm to get the job done.
o) Perform self-assigned software tasks.

2.4.3 Software Engineer Responsibilities

Team members have the following responsibilities:

a) Be knowledgeable of your organization's processes and procedures.
b) Participate in weekly team meetings.
c) Support team progress, problem resolution, and action item closure.
d) Ensure standard tools are used and reliable common processes are used and followed. Make suggestions for tool and process improvements to your Software Manager and/or Lead Software Engineer.
e) Work with your Software Manager and Lead Software Engineer as necessary to develop budget and schedule estimates for team activities and present to the team for approval.
f) Identify budget or schedule impacts to your Lead Software Engineer leader and Software Manager.
g) Evaluate new software requirements and provide feedback to your team.
h) Suggest process improvement changes where you have identified the process flow as inadequate.
i) Participate in scheduled peer reviews.
j) Accept and encourage constructive feedback.

2.5 RISK ANALYSIS

Below is a list of potential risks to the project along with a plan to minimize each risk.

a) Potential risk: Hardware is late pushing the hardware/software integration date back

Plan to minimize impact: Allow two weeks of extra time in schedule to account for the possibility that hardware will be received later than expected.

b) Potential risk: Employee turnover

Plan to minimize impact: Have good training plan in place to help new team members get up to speed quickly with a minimum impact on current team members. Encourage open communication where team members can freely report any problems that they are having with their job.

c) Potential risk: Late changes to requirements cause rework

Plan to minimize impact: Plan for 20% changes to requirements during software development. Thoroughly review initial requirements to find any potential requirements problems and to get these problems resolved early. This will help to minimize the number of requirement problems that will come up during later stages of the software development cycle.

d) Potential risk: Problems found during final testing phases

Plan to minimize impact: Anticipate that problems may be found during all phases of testing and allow adequate time in the schedule, prior to the final release date, to correct these problems.

2.6 HARDWARE AND SOFTWARE RESOURCE REQUIREMENTS

Each person involved in managing or developing software for this project will require a personal computer that meets the minimum hardware and software requirements listed below:

- Pentium 3, 550 MHz processor
- Windows 98 Operating System
- 128 MB RAM
- 19 inch monitor
- Zip Drive for backup
- MS Office 2000 Professional
- MS Project (for schedules) – manager and lead engineer only
- Microsoft Visual C++ 6.0

- Microsoft Outlook (for e-mail)
- Netscape Communicator (Version 4.61 or later)
- Norton AntiVirus 2001
- Rational Rose – Object-oriented design tool
- Rational RequistePro - Requirements management tool that utilizes a database to capture traceability information
- Rational SoDA - Software documentation tool that gathers requirements and design information from RequisitePro and Rose to produce MS-Word documents
- High speed internet access
- ZoneAlarm firewall software
- Configuration Management software

The following lab equipment will be need to perform hardware/software integration and testing:

- In Circuit Emulator (for stepping through code to debug problems on target hardware)
- HP 1631A Logic Analyzer
- Digital Multimeter
- Network server for storing files under configuration control

2.7 WORK BREAKDOWN

See Section 3, Project Schedule.

2.8 PROJECT SCHEDULE

See Section 3, Project Schedule.

2.9 QUALITY

See the Software Quality Assurance Plan in Referenced Documents list for information on the role of Quality Assurance on this project.

2.10 VERIFICATION AND VALIDATION

The project software will be verified against all software requirements to ensure that each requirement has been fully satisfied. See the Verification and Validation Plan in Referenced Documents list for additional information on the how the Pulsar Microwave Oven software will be verified and validated.

2.11 CONFIGURATION MANAGEMENT

See the Software Configuration Management Plan in Referenced Documents list for information on the configuration management system being used on this project.

2.12 MAINTAINABILITY

See the Maintenance Plan in Referenced Documents list for information on the how this software will be maintained.

2.13 STAFF DEVELOPMENT

Persons joining team will be expected to have a minimum of two years of C++ programming experience.

Training classes will be available on specific CASE tools being used during software development.

New employee training will be used to familiarize persons who are new to the project with the project's coding standards, code walk-through process, etc.

For additional information see the Staff Development Plan document in Referenced Documents list.

3. PROJECT SCHEDULE

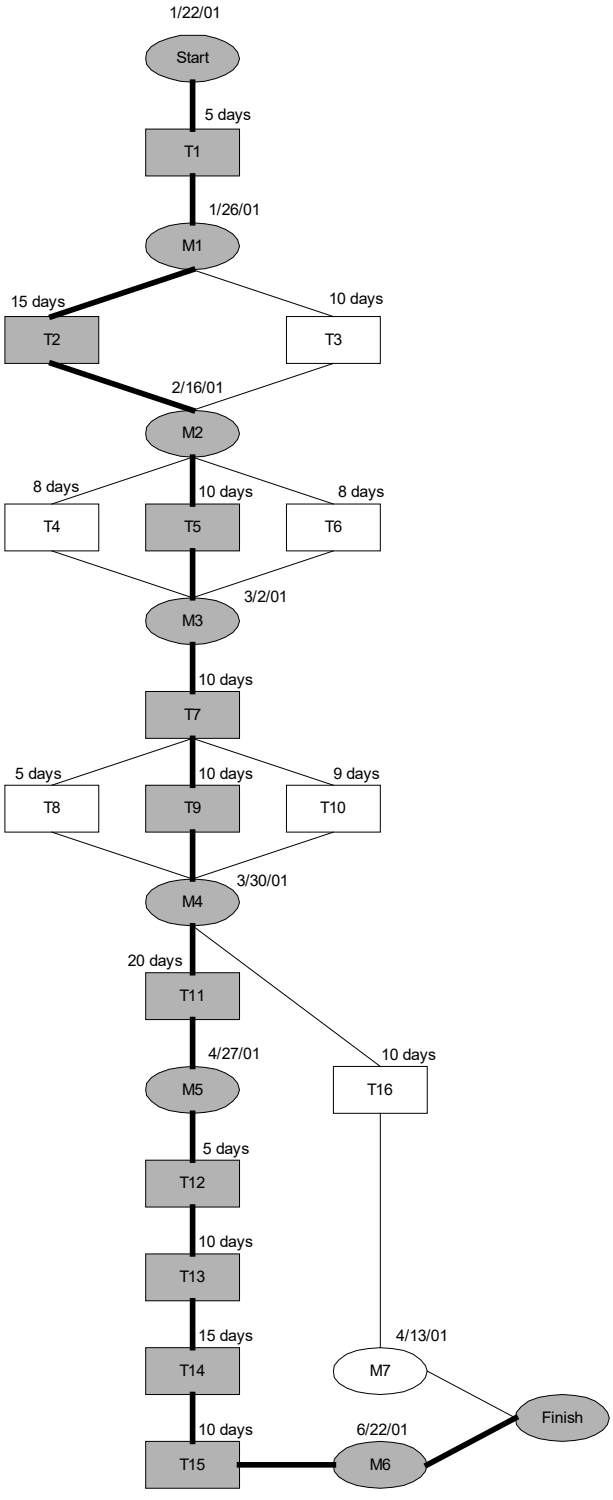
3.1 TASK DURATION AND DEPENDENCIES

Task	Description	Duration in Work Days	Dependent Upon
T1	Concept document	5 days	None
T2	Project Plan	15 days	T1
T3	Project Schedule	10 days	T1
T4	Requirements Analysis	8 days	T1
T5	Requirements Definition	15 days	T1
T6	System Specification	8 days	T1
T7	Requirements Validation	10 days	T4, T5, T6
T8	Architectural Design	5 days	T7
T9	Interface Design	10 days	T7
T10	Detailed Design	9 days	T7
T11	Coding	20 days	T8, T9, T10
T12	Unit Testing	5 days	T11
T13	Integration Testing	10 days	T11, T12
T14	System Testing	15 days	T13
T15	Acceptance Testing	10 days	T14
T16	User Manual	10 days	T8, T9, T10

3.2 ALLOCATION OF PEOPLE TO TASKS

Task	Description	Employee Initials
Task 1	Concept document	AL, DD
Task 2	Project Plan	JG
Task 3	Project Schedule	DD
Task 4	Requirements Analysis	AL, JG
Task 5	Requirements Definition	AL, DD, JG
Task 6	System Specification	AL, DD, JG
Task 7	Requirements Validation	AL, DD
Task 8	Architectural Design	AL, JG
Task 9	Interface Design	AL, DD, JG
Task 10	Detailed Design	AL, DD, JG
Task 11	Coding	AL, BS, DD, JG
Task 12	Unit Testing	AL, BS, DD, JG
Task 13	Integration Testing	AL, BS, DD, JG
Task 14	System Testing	AL, BS, DD, JG
Task 15	Acceptance Testing	AL, BS, DD, JG
Task 16	User Manual	AL, DD, JG

3.3 ACTIVITY NETWORK AND MILESTONE DESCRIPTION



Shaded path is Critical Path
Critical path length is 110 working days

Milestone M1:

A concept document will be generated. Completion of this document indicates that the milestone has been reached.

Milestone M2:

A project plan and project schedule will be generated. Completion of these two items indicates that the milestone has been reached.

Milestone M3:

A Software Requirements Specification will be generated. Completion of this document indicates that the milestone has been reached.

Milestone M4:

A Software Design Document will be generated. Completion of this document indicates that the milestone has been reached.

Milestone M5:

A Fully Functional System Software will be generated. Completion of this software indicates that the milestone has been reached.

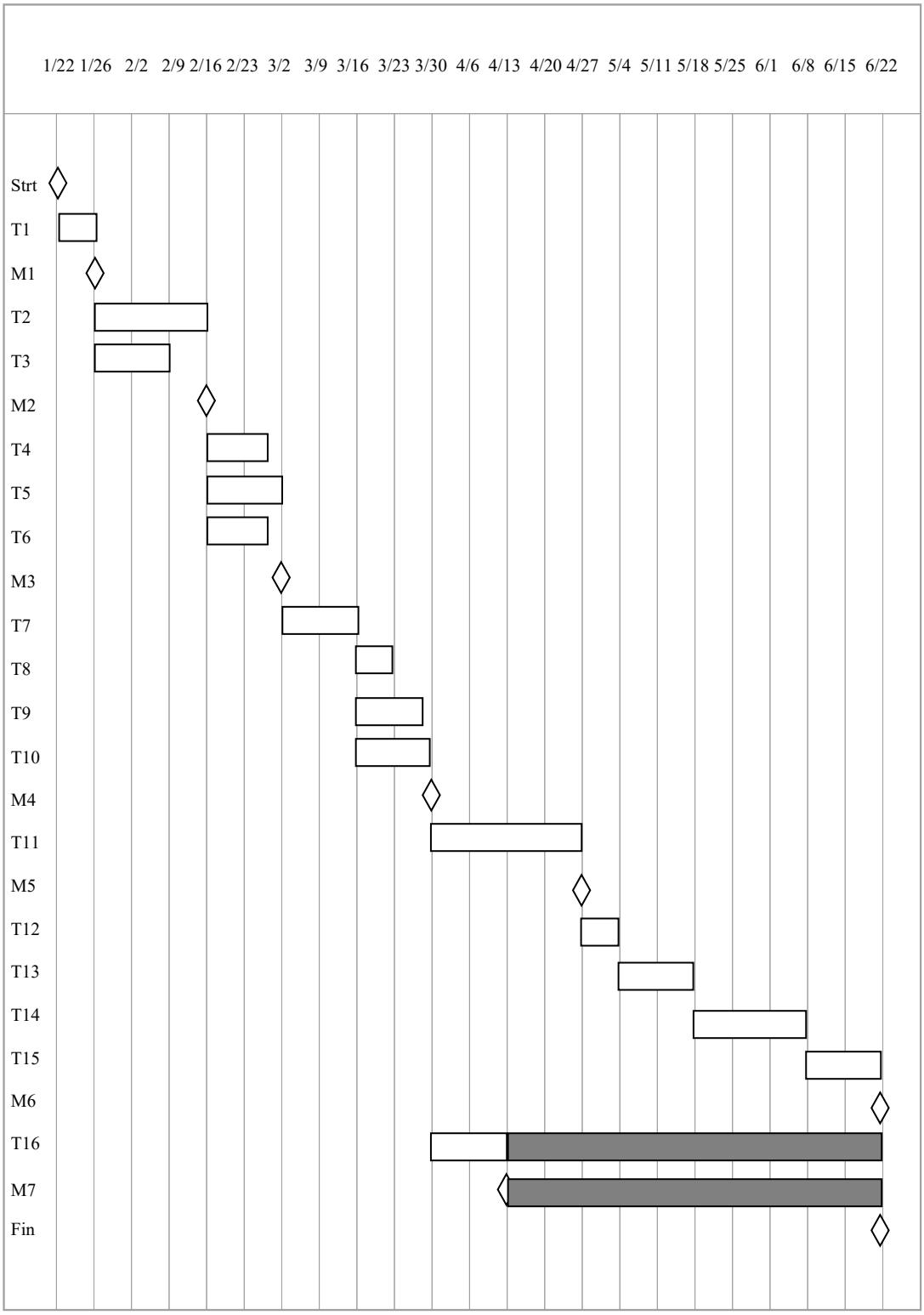
Milestone M6:

An Acceptance Test Report will be generated. Completion of this test indicates that the milestone has been reached.

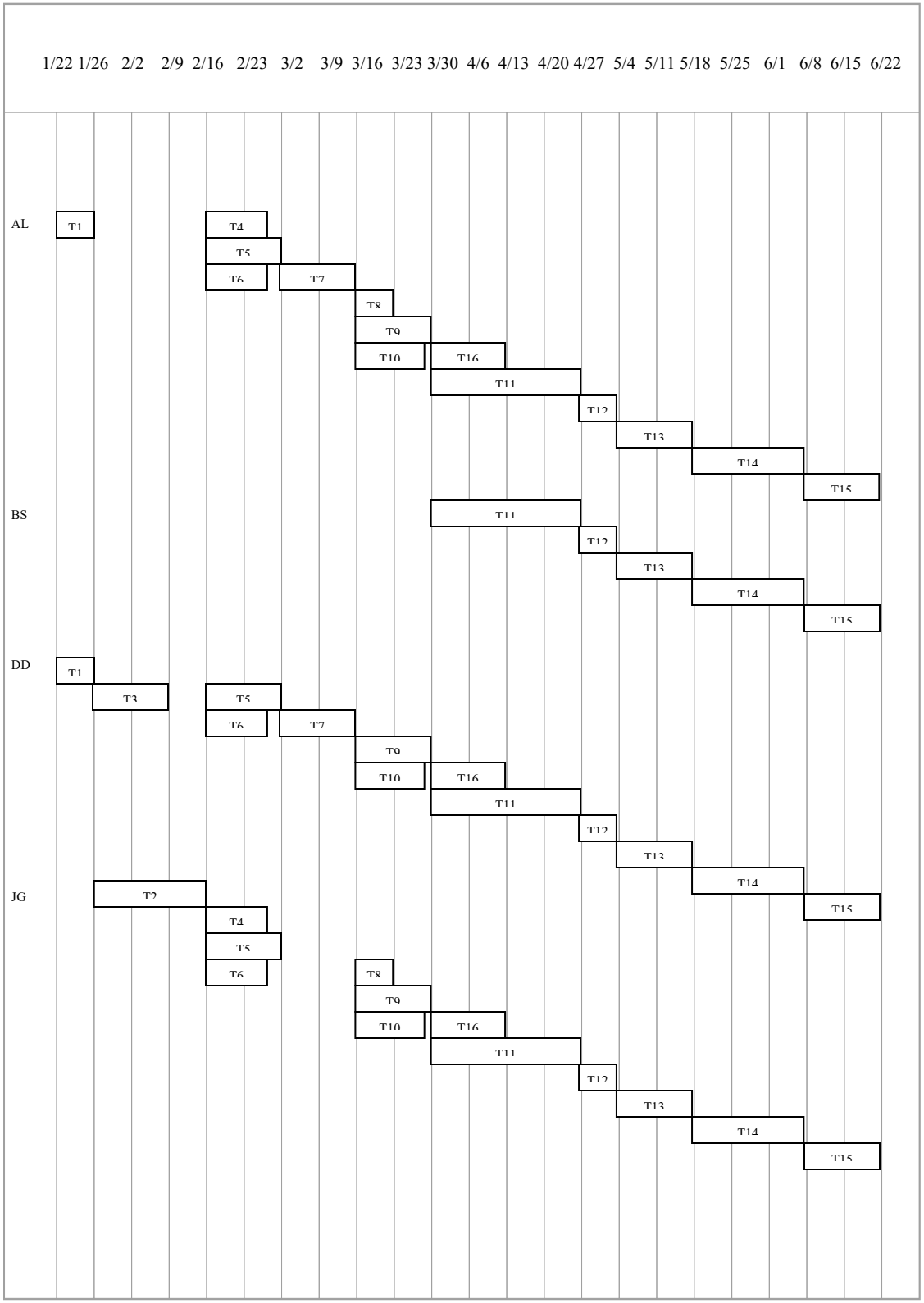
Milestone M7:

A User's Manual will be generated. Completion of this manual indicates that the milestone has been reached.

3.4 ACTIVITY BAR CHART



3.5 STAFF ALLOCATION VERSUS TIME



4. REQUIREMENTS ANALYSIS

4.1 INTRODUCTION

4.1.1 General Requirements Analysis Information

The requirements analysis information in this section is a summary of information located on the following reference URL:
Reference: http://www.cs.binghamton.edu/~mcnair/cs345/present8/sld001.htm

Requirement Analysis bridges the gap between System Engineering and Software Design. The focus in on "what" the system being developed is supposed to do not "how" it will be implemented. Requirements analysis allows system engineers to specify both software function and performance. It provides software engineers with a method for building models of the data, both functional and behavioral. It also provides software designers with models that can be translated into different designs.

Software Requirements Analysis can be broken into the following five parts:
1) Problem Recognition
2) Evaluation and Synthesis
3) Modeling
4) Specification
5) Review

Problem recognition begins by asking questions about the problem without focusing on the solution. The focus in on the customer's overall goals as well as the benefits that can be obtained. It is recommended that the software developers meet with the customer to get a better understanding of the problem and reduce ambiguity.

Once the problem is well defined, it is time to propose and evaluate elements of the solution and negotiate different approaches to solving the problem. A list of constraints and performance criteria should be made.

Models aid the software engineer in understanding the information, function, and behavior of the system. They provide the designers with representations of the software that will make the implementation process easier. There are two types of models: functional models and behavioral models. Creating a functional model begins by focusing on problem specific functions. Iterations are made until all system functionality is represented. Behavioral models on the other hand represent the states of the software and events (internal or external) which cause changes in the state of the software.

The Software Requirements Specification should separate functionality from implementation. It should define the desired behavior and establish the context in which the software operates. It should also define the environment in which the system operates. The specification must be tolerant of incompleteness and be easily updated and changed.

After the Software Requirements Specification is written both the software developer and the customer review it. Once the review is complete, it is signed off by both the customer and the developer. The specification then becomes a "contract" for software development. If changes need to be made, the customer needs to realize that it is after-the-fact and the changes desired might increase the cost and change the project schedule.

4.1.2 Link to Service Usage Model Information

A Service Usage Model was chosen for section 4.6.

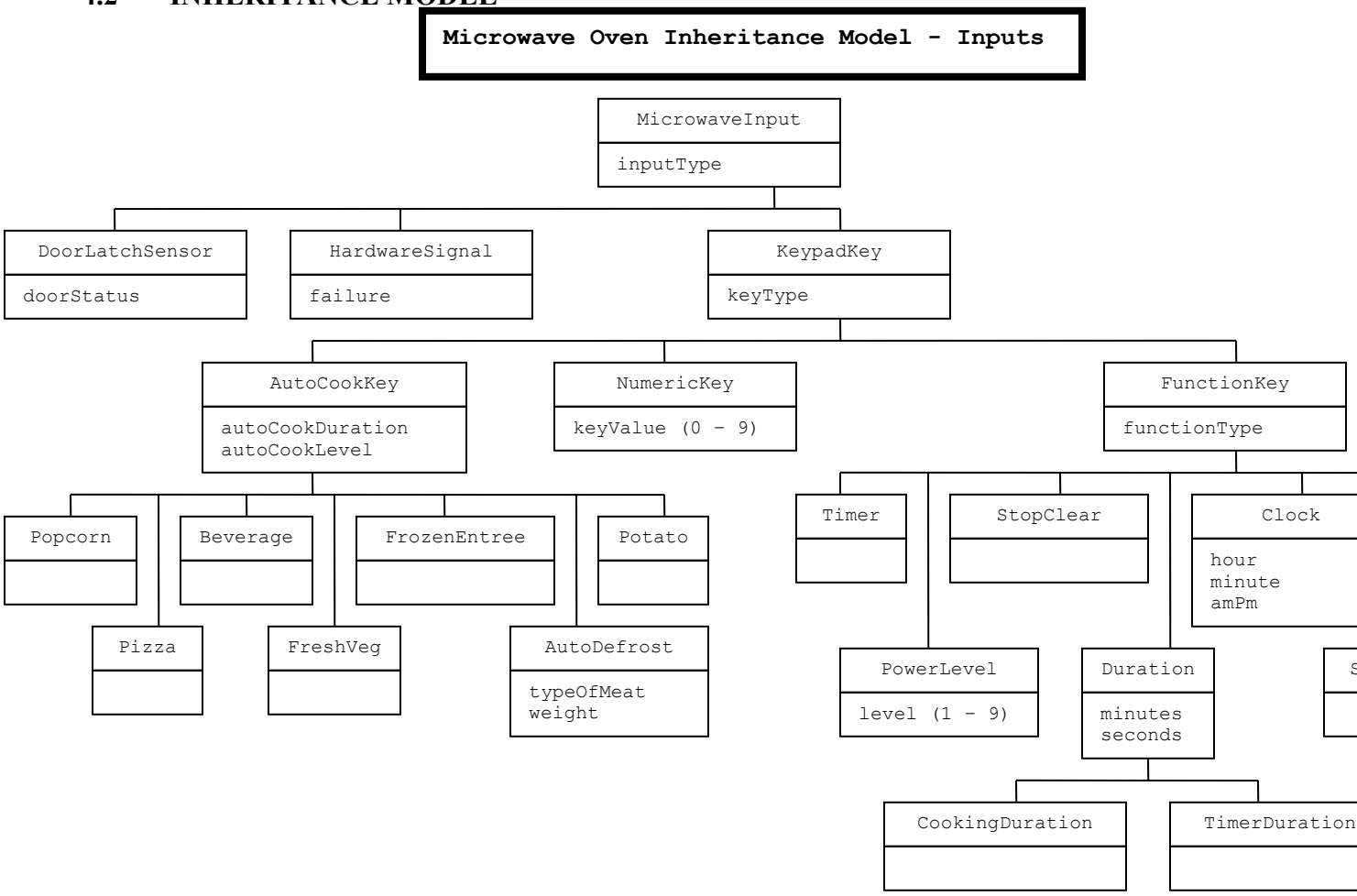
Service usage models supplement object models by showing how one type of class can use the services provided by other types of classes. These models are also useful in recording how external entities use the services provided by classes. The concept of terminators can be borrowed from data flow models and applied to service usage models.

Hierarchical models show object classes and the services associated with each object. They do not provide any information about how the object classes use the services provided by other classes. A service usage model shows how one class is related to other classes through the operations.

The hierarchical models describe the static structure of the system but they do not show how objects interact when the system is executing. It is useful to complement the hierarchical models with a service usage model that will show which services are used by the objects during execution. In some systems there are so many interactions that it is impractical to produce a composite diagram that shows all interactions. In these cases, designers must make their own judgement about which objects and interactions need to be documented. After identifying the object hierarchy and the object interactions, the object interfaces can be defined.

In summary, service usage models provide software designers with a different picture of the system being developed than the other types of models. It is important for the software designers to see the system from various viewpoints to fully understand the system prior to the implementation and coding phase of the program. The better the designers understand the system, and how the objects will interact with each other during execution, the fewer the number of errors that will be found late in the software development cycle.

4.2 INHERITANCE MODEL



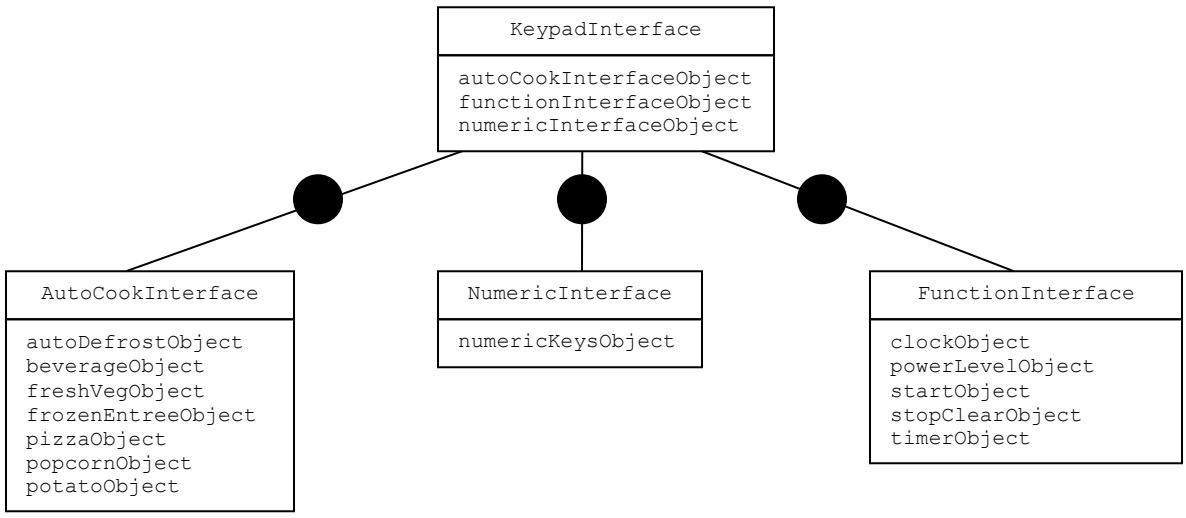
4.2.1 Data Dictionary for Inheritance Model (Figure 4.2)

Term	Description	Type
amPm	AM or PM selection for setting clock.	Attribute
AutoCookKey	Selects Auto-cook preset function of microwave.	Class
autoCookDuration	Preset cooking duration for cooking a specific item.	Attribute
autoCookLevel	Preset cooking power level for cooking a specific item.	Attribute
AutoDefrost	Automatically sets the heating time and power level for defrosting items.	Class
Beverage	Automatically sets the cooking time and power level for hot beverages.	Class
Clock	Selects clock function of microwave.	Class
CookingDuration	Time duration for cooking.	Class
DoorLatchSensor	Senses if door is opened or closed.	Class
doorStatus	Status indication of door. Either microwave door closed indication or microwave door opened indication.	Attribute
Duration	Time duration for cooking or for running a timer.	Class
failure	Hardware failure indication.	Attribute
FreshVeg	Automatically sets the cooking time and power level for fresh vegetables.	Class
FrozenEntree	Automatically sets the cooking time and power level for frozen entrees.	Class
FunctionKey	Selects operational function of microwave.	Class
functionType	Type of function being processed.	Attribute
HardwareSignal	Senses failure conditions of the hardware components.	Class
hour	Hour component of current time of day (in Hours:Minutes). Used for setting the clock.	Attribute
inputType	Type of input data being processed.	Attribute
KeypadKey	Keypad key / button that enables data input and function selection.	Class
keyType	Type of keypad key / button pressed.	Attribute
keyValue	Integer value associated with the numeric key.	Attribute
level	Power level value with a range of 1 to 9.	Attribute
MicrowaveInput	Root input class for the microwave oven.	Class
minutes	Minutes component of initial timer value (Minutes:Seconds).	Attribute
NumericKey	Inputs integer data to microwave oven system.	Class
Pizza	Automatically sets the cooking time and power level for pizza.	Class

Popcorn	Automatically sets the cooking time and power level for popcorn.	Class
Potato	Automatically sets the cooking time and power level for potatoes.	Class
PowerLevel	Allows operator to set desired power level.	Class
seconds	Seconds component of initial timer value (Minutes:Seconds).	Attribute
Start	Start button starts the cooking operation.	Class
StopClear	Stop/Clear button stops the cooking operation.	Class
Timer	Selects timer function of microwave.	Class
TimerDuration	Specifies how long the timer is to run before sounding audible alarm.	Class
typeOfMeat	Type of meat to be defrosted.	Attribute
weight	Weight of meat to be defrosted.	Attribute

4.3 STRUCTURAL/COMPOSITION MODEL

Microwave Oven Structural/Composition Model – Keypad Interface



4.3.1 Data Dictionary for Structural/Composition Model (Figure 4.3)

Term	Description	Type
AutoCookInterface	Class that has the following objects all of type AutoCookPresetKey: autoDefrostObject, beverageObject, freshVegObject, frozenEntreeObject, popcornObject,	Class

	potatoObject, and pizzaObject. This class implements the auto-cook interface.	
autoCookInterfaceObject	The instantiation of the AutoCookInterface type.	Attribute
autoDefrostObject	Automatically sets the heating time and power level for defrosting items.	Attribute
beverageObject	Automatically sets the cooking time and power level for hot beverages.	Attribute
clockObject	Selects clock function of microwave.	Attribute
freshVegObject	Automatically sets the cooking time and power level for cooking fresh vegetables.	Attribute
frozenEntreeObject	Automatically sets the cooking time and power level for cooking frozen entrees.	Attribute
FunctionInterface	Class that has the following objects all of type FunctionKey: clockObject, powerLevelObject, startObject, stopClearObject, and timerObject. This class implements the function keys interface.	Class
KeypadInterface	Class that has the following objects: autoCookInterfaceObject : AutoCookInterface, functionInterfaceObject : FunctionInterface, and numericInterfaceObject : NumericInterface. This class implements the numeric keys interface.	Class
NumericInterface	Class that has the following objects: numericKeysObject : NumericKey. This class implements the numeric keys interface.	Class
numericInterfaceObject	The instantiation of the NumericInterface type.	Attribute
numericKeysObject	Inputs integer data to microwave oven system.	Attribute
pizzaObject	Automatically sets the cooking time and power level for cooking pizza.	Attribute
popcornObject	Automatically sets the cooking time and power level for popping popcorn.	Attribute
potatoObject	Automatically sets the cooking time and power level for cooking potatoes.	Attribute
powerLevelObject	Allows operator to set desired power level.	Attribute
startObject	Start button starts the cooking operation.	Attribute
stopClearObject	Stop/Clear button stops the cooking operation.	Attribute
timerObject	Selects timer function of microwave.	Attribute

4.4 DATA-FLOW MODEL

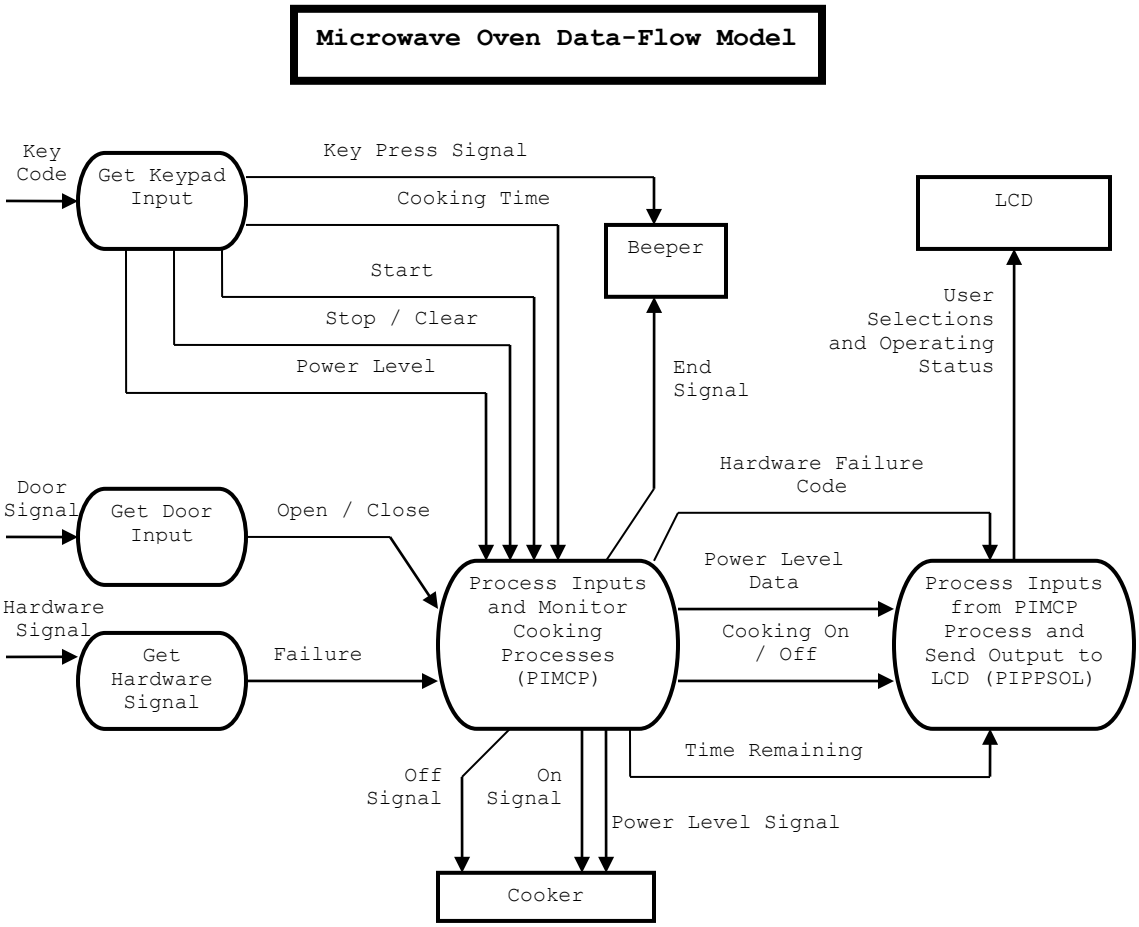


Figure 4.4 Data Flow Model

4.4.1 Data Dictionary for Data Flow Model (Figure 4.4)

Term	Description	Type
Beeper	Produces a tone when keys are pressed or when a heating process has finished.	System Component
Cooker	This device controls the radiation emitter and accepts control signals from the PIMCP.	System Component
Cooking On / Off	Cooking status data that is sent as output from the PIMCP process and becomes input to PIPPSOL process for conversion into LCD format. It indicates the current cooking status of the oven which is either on or off.	Intermediate Output
Cooking Time	A partially processed input that specifies the amount of time a food item will be heated. This is set either manually by the user or automatically when the user selects a preset function.	Processed Input
Door Signal	Binary code from the door sensor.	Input
End Signal	Binary code sent to the beeper component that commands the beeper to produce the audible alarm indicating the end of the cooking cycle.	Output
Failure	A partially processed input that indicates a hardware failure.	Processed Input
Get Door Input	The microwave oven system accepts signals from the oven door sensors which are processed to determine whether the microwave door is opened or closed.	Process
Get Keypad Input	The microwave oven system accepts signals from the keypad which are processed to determine the user's selections.	Process
Get Hardware Signal	The microwave oven system accepts hardware signals from various hardware components which are processed to produce failure codes for display.	Process
Hardware Failure Code	This code data is derived from hardware component failure signals and is sent to the PIPPSOL for display processing.	Intermediate Output
Hardware Signal	Binary code from hardware component.	Input
Key Press Signal	Binary code sent to the Beeper that indicates a keypad key was pressed.	Output
Key Code	Binary code from the keypad.	Input
LCD	The LCD (Liquid Crystal Display) displays input and output messages.	System Component
Off Signal	Binary code sent to the cooker component that commands the cooker component to turn off.	Output
On Signal	Binary code sent to the cooker component that commands the cooker component to turn on.	Output
Open / Close	A partially processed user input that indicates the current status of the microwave door as to whether it is open or closed.	Processed Input
PIMCP	The Process Inputs and Monitor Cooking Processes (PIMCP) process accepts and processes input from the keypad, door sensor,	Process

	and cooking device input processing processes. The PIMCP also controls and monitors cooking processes.	
PIPPSOL	The Process Inputs from PIMCP Process and Send Ouput to LCD (PIPPSOL) process accepts and processes input from the PIMCP process and sends output data to the LCD for display.	Process
Power Level	A partially processed input that specifies the power level selected by the user. Power Level is set manually by the user or through a preset function.	Input
Power Level Data	Power level data is sent as output from the PIMCP process and becomes input to PIPPSOL process for conversion into LCD format.	Intermediate Output
Power Level Signal	Binary code sent to the cooker component that commands the cooker component to cook at the specified level of power.	Output
Start	A partially processed user input that signals that a heating process should begin.	Processed Input
Stop/Clear	A partially processed user input that signals that a heating process should be stopped and/or a cooking time should be cleared from memory.	Processed Input
Time Remaining	Time remaining data is sent as output from the PIMCP process and becomes input to PIPPSOL process for conversion into LCD format. This data is the amount of time remaining for the currently active cooking cycle.	Intermediate Output
User Selections and Operating Status	The User Selections and Operating Status data packet is sent to the LCD component for display as an output from the PIPPSOL process.	Output

4.5 SEMANTIC MODEL

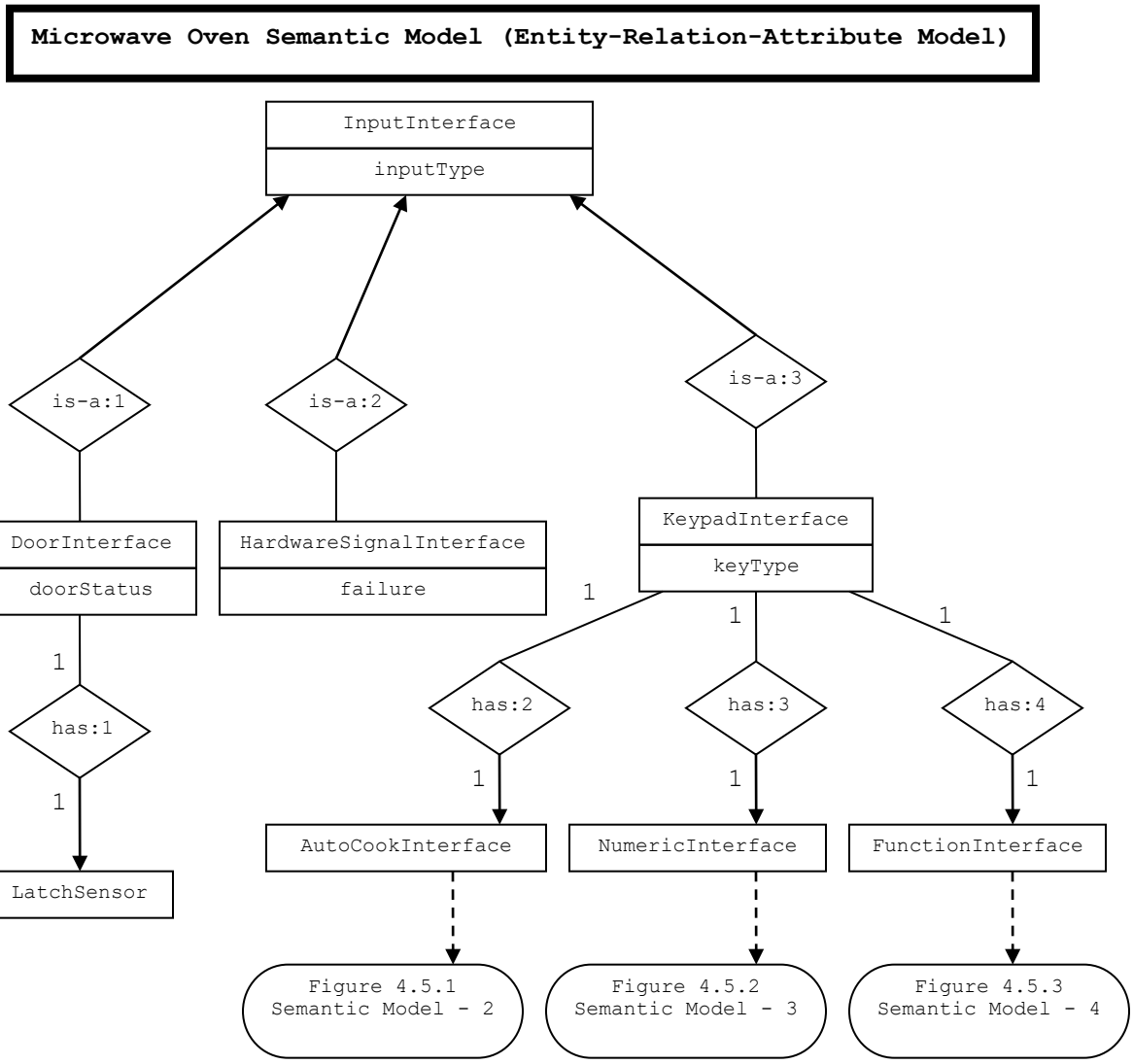


Figure 4.5 Semantic Model

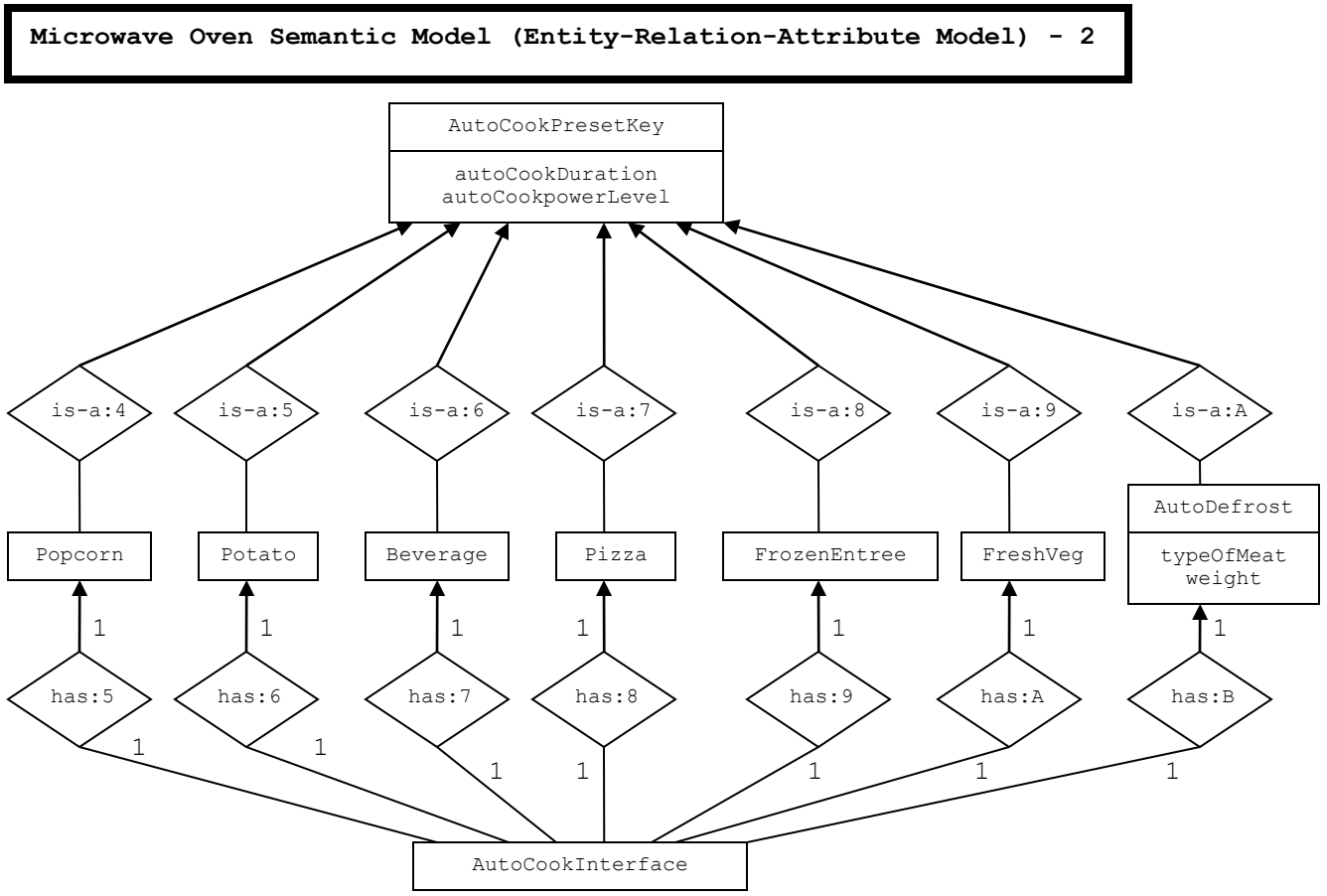


Figure 4.5.1 Semantic Model (continued)

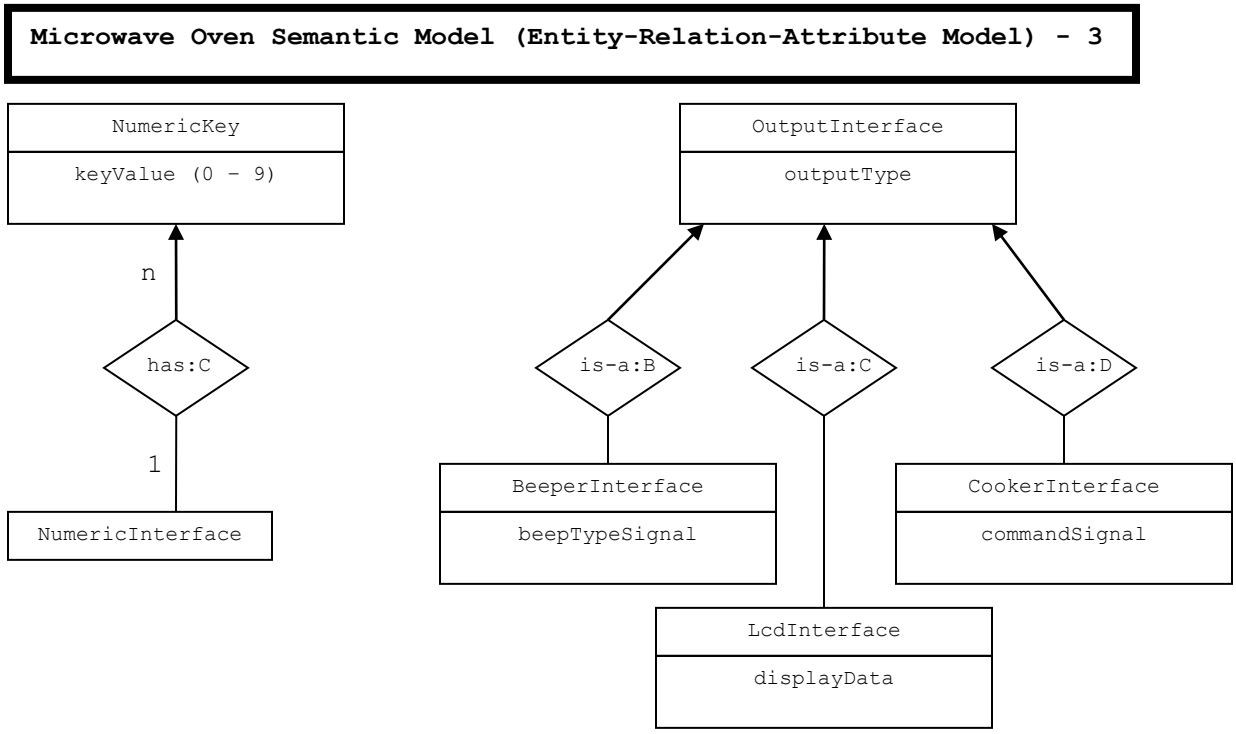


Figure 4.5.2 Semantic Model (continued)

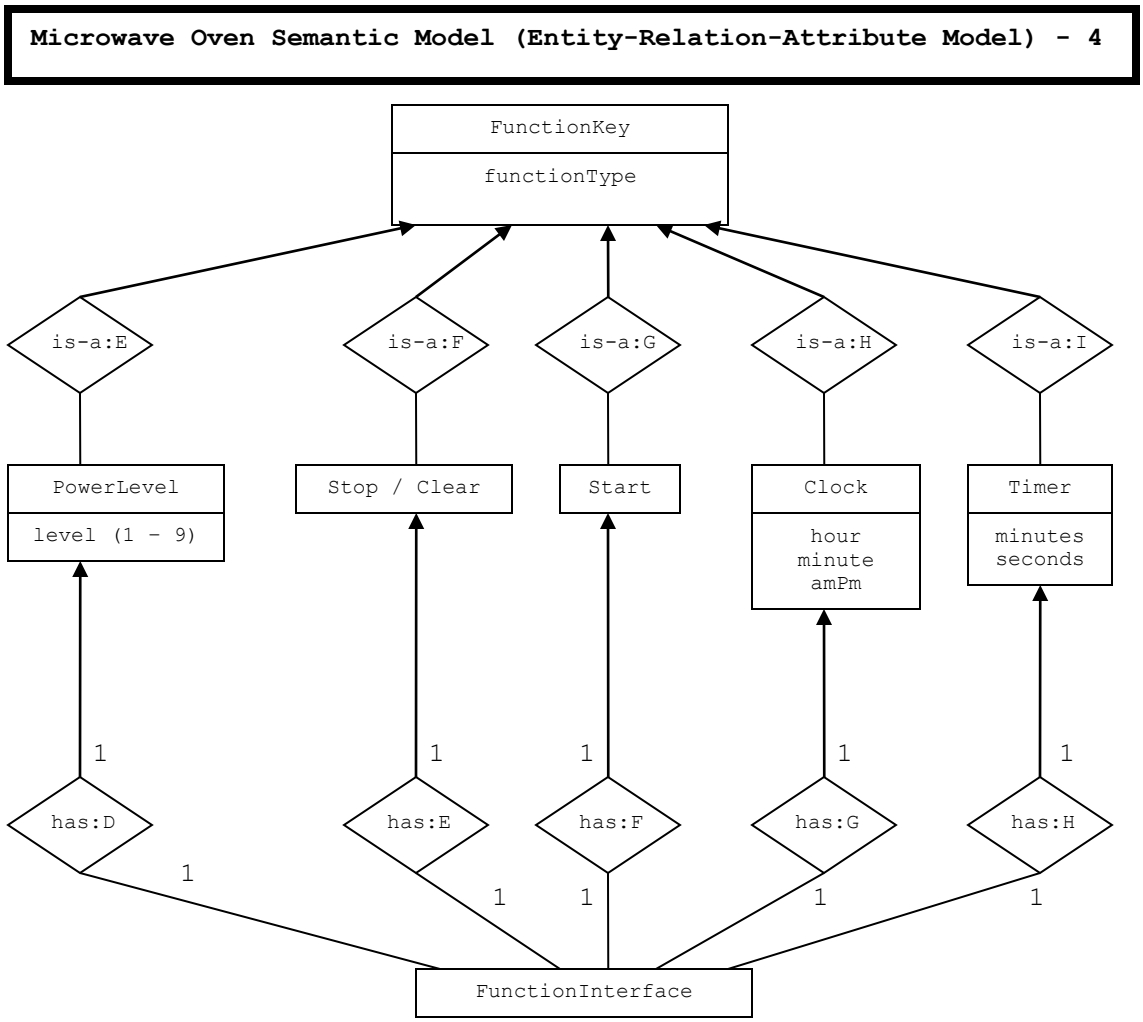


Figure 4.5.3 Semantic Model (continued)

4.5.1 Data Dictionary for Semantic Model (Figure 4.5)

Term	Description	Type
amPm	Contains the time-of-day integer antemeridiem / postmeridiem value.	Attribute
autoCookDuration	Contains the time period value for the amount of time the cooker component is to cook.	Attribute
AutoCookInterface	This interface contains all of the AutoCookPresetKey entities.	Entity
autoCookPowerLevel	Contains the power level of the cooker component.	Attribute
AutoCookPresetKey	Parent entity to the following AutoCookPresetKey-derived entities: Popcorn, Potato, Beverage, Pizza, FrozenEntree, FreshVeg. Manufacturer defined presets for heating common food items.	Entity
AutoDefrost	Derived from the AutoCookPresetKey entity. Key used to set power level and time for defrosting frozen food items.	Entity
BeeperInterface	Derives from OututInterface and provides communication with beeper component.	Entity
beepTypeSignal	Contains the style of audible alarm to be sounded.	Attribute
Beverage	Derived from the AutoCookPresetKey entity. Automatically sets the heating time and power level for hot beverages.	Entity
Clock	Derived from the FunctionKey entity. Key used in conjunction with numeric keys to set the clock.	Entity
commandSignal	Contains the specified command for the cooker component.	Attribute
CookerInterface	Derives from OutputInterface and provides communication with the cooker component.	Entity
displayData	Contains the data to be displayed on the LCD screen.	Attribute
DoorInterface	Derives from InputInterface and contains data from the door latch sensor	Entity
doorStatus	Status input from door latch sensor.	Attribute
failure	Failure indication from cooker component (if failure occurs).	Attribute
FreshVeg	Derived from the AutoCookPresetKey entity. Automatically sets the heating time and power level for fresh vegetables.	Entity
FrozenEntrée	Derived from the AutoCookPresetKey entity. Automatically sets the heating time and power level for frozen entrees.	Entity
FunctionInterface	This interface contains all of the FunctionKey entities.	Entity
FunctionKey	Part of the FunctionInterface. Allows execution of functions such as PowerLevel, Program,	Entity

	AutoDefrost, Stop/Clear, Start, Clock, and Timer.	
functionType	Contains the type of function.	Attribute
HardwareSignalInterface	Derives from InputInterface and provides communication with hardware components.	Entity
has:1	A single DoorInterface entity has a single LatchSensor entity.	Relation
has:2 – has:4	A single KeypadInterface entity has a single instance of each of the following entities: AutoCookInterface, NumericInterface, and FunctionInterface.	Relation
has:5 – has:B	A single AutoCookInterface entity has a single instace of each of the following entities: Popcorn, Potato, Beverage, Pizza, FrozenEntree, FreshVeg.	Relation
has:C	A single NumericInterface entity has multiple NumericKey entities.	Relation
has:D – has:H	A single FunctionInterface entity has a single instace of each of the following entities: PowerLevel, AutoDefrost, Stop/Clear, Start, Clock, and Timer.	Relation
hour	Contains the time-of-day integer hour value.	Attribute
InputInterface	Parent entity to the following InputInterface-derived entities: DoorInterface, CookerInterface, KeypadInterface.	Entity
inputType	Type of interface data.	Attribute
is-a:1 - is-a:3	The DoorInterface, CookerInterface (input), and KeypadInterface are all entities derived from the InputInterface entity.	Relation
is-a:4 - is-a:A	Popcorn, Potato, Beverage, Pizza, FrozenEntree, and FreshVeg, and AutoDefrost are all entities derived from the AutoCookPresetKey entity.	Relation
is-a:B - is-a:D	BeeperInterface, LcdInterface, and CookerOutputInterface are all entities derived from the OutputInterface entity.	Relation
is-a:E - is-a:I	PowerLevel, AutoDefrost, Stop/Clear, Start, Clock, and Timer are all entities derived from the FunctionKey entity.	Relation
KeypadInterface	Derives from InputInterface. KeypadInterface has a single instance of each of the following entities: AutoCookInterface, NumericInterface, FunctionInterface.	Entity
keyType	Contains the type of interface to which a given key belongs.	Attribute
keyValue	Contains the integer value of the numeric key.	Attribute
LatchSensor	Senses whether or not door is open.	Entity
LcdInterface	Derives from OutputInterface and provides communication with LCD.	Entity
level	Contains the integer value of the selected level of power for the cooker component.	Attribute

minute	Contains the time-of-day integer minute value.	Attribute
minutes	Contains the timer integer minutes value that specifies the number of minutes.	Attribute
NumericInterface	This interface contains all of the NumericKey entities.	Entity
NumericKey	Part of the NumericInterface. Keys are numbered 0-9 for manually entering numeric values used for setting the clock, the timer, the power level, and the cooking time.	Entity
OutputInterface	Parent entity to the following OutputInterface-derived entities: BeeperInterface, LcdInterface, CookerInterface.	Entity
outputType	Type of interface data.	Attribute
Pizza	Derived from the AutoCookPresetKey entity. Automatically sets the heating time and power level for pizza.	Entity
Popcorn	Derived from the AutoCookPresetKey entity. Automatically sets the heating time and power level for microwave popcorn	Entity
Potato	Derived from the AutoCookPresetKey entity. Automatically sets the heating time and power level for one potato.	Entity
PowerLevel	Derived from the FunctionKey entity. Key used in conjunction with numeric keys to change power level.	Entity
seconds	Contains the timer integer seconds value that specifies the number of seconds.	Attribute
Start	Derived from the FunctionKey entity. Key used to begin a cooking process.	Entity
Stop/Clear	Derived from the FunctionKey entity. Key used to stop a cooking process and/or clear a heating time.	Entity
Timer	Derived from the FunctionKey entity. Key used in conjunction with numeric keys to set the timer.	Entity
typeOfMeat	Type of meat to be defrosted.	Attribute
weight	Weight of meat to be defrosted.	Attribute

4.6 SERVICE USAGE MODEL

Service usage models describe the services used by objects in the system. In this case, the objects would include the user as well as the food items being prepared by the user. The services include the features and options offered by the microwave unit.

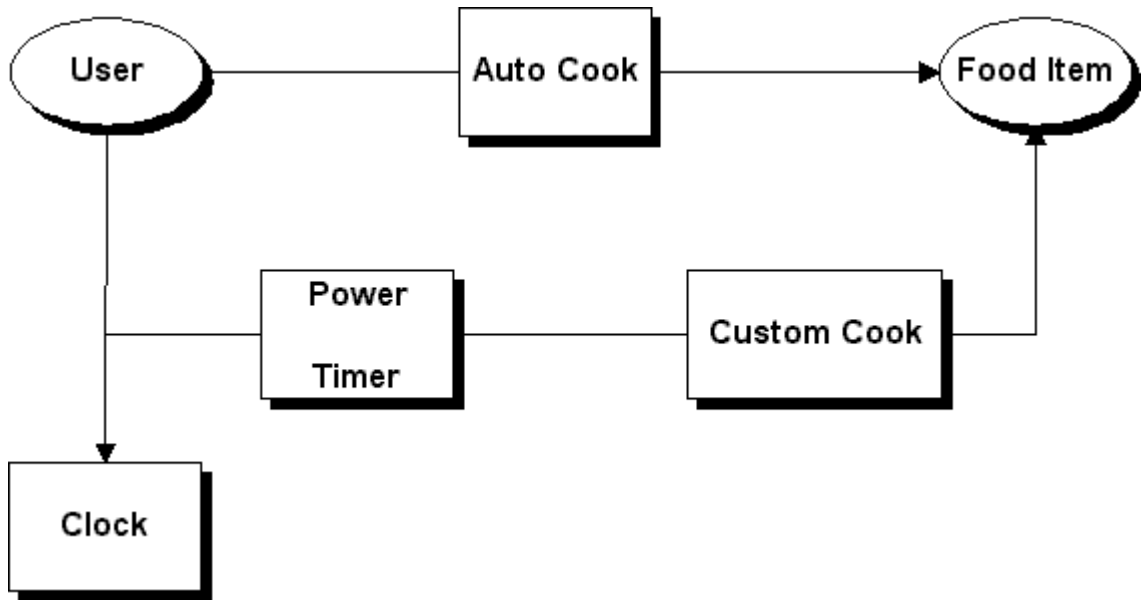


Figure 4.6 Service Usage Model

4.6.1 Data Dictionary for Service Usage Model (Figure 4.6)

Term	Description	Type
User	Individual using microwave.	Object
Food Item	Item to be prepared by user.	Object
Auto Cook	Auto Cook service includes auto cook preset functions.	Service
Power	Allows user to manually set power level.	Service
Timer	Allows user to manually set the microwave timer.	Service
Custom Cook	Custom Cook service includes manual power and timer functions. Allows user to heat food items not included in auto cook presets.	Service
Clock	Allows user to manually set the time displayed on the microwave clock.	Service

5 REQUIREMENTS DEFINITION

5.1 INTRODUCTION

5.1.1 Verifiable Requirements

What does it mean for a requirement to be verifiable?

A requirement is verifiable if a test can be written and executed to prove that the delivered system meets the requirement.

5.1.2 High-quality Requirements

What checks did you use to ensure that all your requirements are high-quality requirements?

1) Does the requirement have a unique number?
2) Is the requirement an amalgamation of several requirements (could the requirement be rewritten to be several separate requirements)?
3) Is the requirement clearly stated?
4) Is the requirement verifiable (could a test be written to verify the requirement)?

5.2 OVERALL SYSTEM NON FUNCTIONAL REQUIREMENTS

5.2.1 Programming Language

5.2.1.1 C++ shall be used to implement the microwave control software.

Rationale: C++ is a well known, proven language for implementing software designs

5.2.2 Operational Temperature Range

5.2.2.1 The microwave shall operate properly from 0 degrees Celsius to 45 degrees Celsius.

Rationale: The microwave should function properly within the range of temperatures found in a majority of homes and business offices.

5.2.3 Operational Voltage Range

5.2.3.1 The microwave shall function properly with an input voltage of 120 VAC +/- 10 VAC.

Rationale: The microwave should operate on common AC voltages found in the United States of America and in Canada.

5.2.4 Reliability

5.2.4.1 The microwave shall have a minimum mean time between failure rate of 4,000 hours.

Rationale: High reliability is important to ensure customer satisfaction.

5.3 FUNCTIONAL REQUIREMENTS

5.3.1 General Requirements

5.3.1.1 Each time a keypad button is pressed a tone shall sound.

Rationale: The operator should be provided with feedback that the software sensed that a pad had been pressed.

Non Functional Requirement: The tone shall sound for 1 second, +/- 200 msec.

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.1.1 of the Requirement Specification>

5.3.1.2 A tone shall sound each time the door is opened or closed.

<See Section 1.1.6 of the Concept Document>

Rationale: The tone indicates that the microwave door has been opened or closed.

Non Functional Requirement: The tone shall sound for 2 second, +/- 200 msec.

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.1.2 of the Requirement Specification>

5.3.1.3 When the microwave is powered on it shall default to the clock function.

Rationale: The clock function is the desired default function.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.1.3 of the Requirement Specification>

5.3.1.4 The microwave software shall support the following functions as described in more detail in the requirements that follow:

a) Clock
b) Timer
c) Auto Cook Presets
d) Auto Defrost
e) Timed Cooking

<See Section 1.1.6 of the Concept Document>

Rationale: These are the desired functions for the microwave oven being developed.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.1.4 of the Requirement Specification>

5.3.1.5 The active microwave function shall be identified on the microwave LCD screen.

Rationale: The operator should be about to easily see which function is currently active by looking on the LCD screen (e.g., is the timer value or cooking time value currently being displayed).

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.1.5 of the Requirement Specification>

5.3.2 Clock

5.3.2.1 The Clock function of the microwave shall be activated when the Clock keypad button is pressed.

<See Sections 1.1.6 and 1.2.2 of the Concept Document>

Rationale: The purpose of the clock button is to activate the clock function of the microwave oven.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.2.1 of the Requirement Specification>

5.3.2.2 The default clock value at power up should be 12:00.

Rationale: 12:00 is the desired default clock time. The operator can adjust the time to the correct value from this setting.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.2.2 of the Requirement Specification>

5.3.2.3 The clock shall be operator settable, in one-minute increments, from 12:00 to 11:59.

Rationale: The operator should be able to set the clock to any minute of the day.

Non Functional Requirement: The Clock time shall be displayed in twelve (12) hour format.

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.2.3 of the Requirement Specification>

5.3.2.4 When the clock function is active, the current time shall be displayed on the microwave LCD screen.

Rationale: The operator should be able to clearly see the current time-of-day when no other microwave function is active.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.2.4 of the Requirement Specification>

5.3.2.5 The initial clock value shall be settable by using the keys 0 through 9 on the microwave touch pad.

<See Section 1.2.2 of the Concept Document>

Rationale: The keys 0 through 9 make it easy for the operator to set the clock value to the desired start value.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes

4) Verifiable? Yes

Checks failed: None

<See Table 6.2.5 of the Requirement Specification>

5.3.2.6 When the clock function is active, and the Start button is pressed, the clock shall begin incrementing the current time value once a minute.

<See Section 1.2.2 of the Concept Document>

Rationale: The clock should begin measuring time after the Start button is pressed.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.2.6 of the Requirement Specification>

5.3.2.7 A colon symbol (":") shall be displayed between the hours and minutes numerals on the display.

Rationale: The colon symbol clearly separates the hours from the minutes in the displayed time value.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.2.7 of the Requirement Specification>

5.3.2.8 When the clock is running, the colon symbol between the hours and minutes numerals on the display shall be alternately turned on and off each second.

Rationale: The flashing colon symbol indicates that the clock is running.

Non Functional Requirement: None

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.2.8 of the Requirement Specification>

5.3.2.9 When the clock is not running, the colon symbol shall not be turned on and off , but shall remain on.

Rationale: After setting the time, the operator should be able to easily see whether or not the clock has been restarted.

Non Functional Requirement: None

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.2.9 of the Requirement Specification>

5.3.3 Timer

5.3.3.1 The Timer function of the microwave shall be activated when the Timer keypad button is pressed.
<See Sections 1.1.6 and 1.2.2 of the Concept Document>

Rationale: The purpose of the Timer button is to activate the Timer function of the microwave oven.

Non Functional Requirement: None

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.3.1 of the Requirements Specification>

5.3.3.2 The default timer value at power up should be 00:00.

Rationale: Assume the operator would want an initial timer value of 00:00.

Non Functional Requirement: None

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.3.2 of the Requirements Specification>

5.3.3.3 The initial timer value shall be settable by using the keys 0 through 9 on the microwave touch pad.
<See Section 1.2.2 of the Concept Document>

Rationale: The keys 0 through 9 make it easy for the operator to set the timer value to the desired start value.

Non Functional Requirement: The range of valid initial timer values shall be from 99 minutes, 99 seconds to 0 minutes, 1 second.

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.3.3 of the Requirements Specification>

5.3.3.4 If the timer value is between 99:99 and 00:01 inclusive and the Start button is pressed, the timer value shall be decremented by one second each second until either the Stop/Clear keypad is pressed or the timer reaches 00:00.
<See Section 1.2.2 of the Concept Document>

Rationale: The display has a resolution of one second so it makes sense to decrement the timer value every second.

Non Functional Requirement: None

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.3.4 of the Requirements Specification>

5.3.3.5 Within 200 msec after the timer decrements from the value of 00:01 to the value 00:00 a buzzer shall sound.

<See Section 1.1.6 of the Concept Document>

Rationale: The buzzer sound will notify the operator that the timer has reached 00:00.

Non Functional Requirement: The buzzer shall sound for three seconds +/- 200 msec.

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.3.5 of the Requirements Specification>

5.3.3.6 While the timer function is active, the current timer value shall be displayed on the microwave's LCD display.

Rationale: The operator should be able to clearly see how much time is left during countdowns.

Non Functional Requirement: None

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.3.6 of the Requirements Specification>

5.3.3.7 If the Stop/Clear button on the microwave keypad is pressed while the timer value is decrementing, the timer value shall stop decrementing and the timer value at the time the Stop/Clear button was pushed shall be displayed.

<See Section 1.2.2 of the Concept Document>

Rationale: The operator should have the capability to stop the timer during the countdown.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.3.7 of the Requirements Specification>

5.3.3.8 If the Timer was stopped during countdown and the Start keypad is pressed the timer value shall once again begin decrementing from the time at which the timer was stopped.

Rationale: The operator should have the option of restarting the timer if desired. Logically pressing the Start button should restart the countdown.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.3.8 of the Requirements Specification>

5.3.3.9 If the Timer was stopped during countdown and the Stop/Clear button is pushed a second time the timer value should immediately be set to 00:00 and the buzzer shall sound to indicate that the time countdown has been canceled.

<See Section 1.2.2 of the Concept Document>

Rationale: It is assumed that if the operator pushes the Stop/Clear button a second time then the operator wants to exit the timer function.

Non Functional Requirement: The buzzer shall sound for 2 seconds, +/- 200 msec.

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.3.9 of the Requirements Specification>

5.3.3.10 The Timer countdown shall not stop if the microwave door is opened during the countdown.

Rationale: The timer is independent of the timed cooking function and therefore there is no safety hazard in allowing the countdown to continue if the microwave door is opened.

Non Functional Requirement: None

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.3.10 of the Requirements Specification>

5.3.4 Auto Cook Presets

5.3.4.1 The microwave shall provide a series of presets for heating common food items. A preset will be included for each of the following: microwave popcorn, baked potatoes, pizza, frozen entrees, fresh vegetables, and liquid beverages.

<See Sections 1.1.6 and 1.2.2 of Concept Document>

Rationale: Providing presets will allow a user to heat common food items more conveniently and in fewer steps.

Non Functional Requirement: All preset keys shall be grouped together on the user interface. All presets shall require only one touch to activate.

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.4.1 of Requirements Specification>

5.3.4.2 Each preset button shall have the capability of having a unique cooking time and power setting associated with it.

Rationale: Each type of preset food may have different cooking time and power settings.

Note: The specific cooking time and power settings for each preset button will be identified in the detailed design document.

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.4.2 of Requirements Specification>

5.3.4.3 When an operator presses the Start button following a selecting an auto cook preset button:

a) the oven shall be turned on at the correct predefined power setting
b) the correct predefined cooking time shall be displayed in the LCD screen
c) the cooking time shall begin decrementing

Rationale: In response to the Start button being pressed, after a preset button has been pushed, the microwave should cook the type of food selected for the correct amount of time at the correct power setting.

Non Functional Requirement: None

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No

3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.4.3 of Requirements Specification>

Note: Requirements 5.3.6.7, 5.3.6.8, 5.3.6.9, 5.3.6.10, and 5.3.6.11 listed below under Timed Cooking also apply to the Auto Cook Preset function.

5.3.5 Auto Defrost

5.3.5.1 The microwave shall provide an Auto Defrost function that will automatically set the defrost time and power level for frozen meat items depending on the weight of the item entered by the user.

Rationale: Providing an Auto Defrost function will allow users to quickly and conveniently defrost frozen meat products, thereby reducing preparation time for entire meals.

Non Functional Requirement: The Auto Defrost key shall be located on the user interface below the bottom row of the numeric keypad.

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.5.1 of Requirements Specification>

5.3.5.2 The Auto Defrost function shall include three settings: Beef, Poultry, and Fish with the capability of providing a unique power level setting for each of these types of meat.

Rationale: Different types of meat require heating under different power levels in order to defrost properly. The process of selecting the type of meat to be defrosted should be made as simple as possible. The process should be simple enough for a first-time user to complete successfully.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.5.2 of Requirements Specification>

5.3.5.3 A menu shall be displayed on the LCD screen to provide the operator with the choice of selecting Beef, Poultry, or Fish.

Rationale: Providing a menu will simplify the process of selecting the type of meat the user wishes to defrost.

Non Functional Requirement: A numerical value shall be included next to each choice on the menu to indicate that the number should be pressed on the keypad to select the type of meat to be defrosted. If a choice is not made in 15 seconds the default setting of poultry shall be selected.

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.5.3 of Requirements Specification>

5.3.5.4 The Auto Defrost function shall require the user to enter the approximate weight of the meat item being defrosted.

Rationale: The defrosting time will depend on the relative mass of the food item being defrosted.

Non Functional Requirement: The weight shall be entered in pounds and ounces.

Inspection results:
Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes
Checks failed: None

<See Table 6.5.4 of Requirements Specification>

5.3.5.5 The user shall be prompted by a message displayed on the LCD to enter the approximate weight of the meat item using the numeric keypad.

Rationale: The user must be prompted for all relevant information in order to make the process as simple as possible.

Non Functional Requirement: The format of the lettering should follow the requirements set out in the General Requirements section.

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.5.5 of Requirements Specification>

5.3.5.6 The Auto Defrost function shall require that the start key be pressed when the user is ready to begin the defrosting operation.

Rationale: Pressing the start key will notify the microwave control software that the user has finished entering the appropriate information and that the defrosting process can begin.

Non Functional Requirement: The user should be prompted by a message on the LCD to press the start key when all the appropriate information has been entered.

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.5.6 of Requirements Specification>

Note: Requirements 5.3.6.7, 5.3.6.8, 5.3.6.9, 5.3.6.10, and 5.3.6.11 listed below under Timed Cooking also apply to the Auto Defrost function.

5.3.6 Timed Cooking

<See Section 1.1.6 of Concept Document>

5.3.6.1 Prior to starting the timed cooking, the operator shall have the capability to choose a power level from 1 to 9 by first pressing the Power Level button followed by one of the pads 1 through 9.

<See Section 1.2.2 of Concept Document>

Rationale: The operator should be able to choose any of the nine available power levels

Assumption: Level 1 will be the lowest power level, Level 2 will be the second lowest power level, etc. such that Level 9 is the highest power level available.

Non Functional Requirement: If a number pad (1 through 9) is not pressed within 15 seconds the default value of Level 9 should be used.

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.1 of Requirements Specification>

5.3.6.2 The default Power setting shall be level 9.

Rationale: It is assumed that a majority of the cooking will be done at the highest power level.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.2 of Requirements Specification>

5.3.6.3 The initial cooking time value shall be settable using the pads 0 through 9 on the microwave touch pad.

Rationale: The keys 0 through 9 make it easy for the operator to set the timer value to the desired start value.

Non Functional Requirement: The valid range of cooking times should be from 99 minutes and 99 seconds to zero minutes and 1 second.

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.3 of Requirements Specification>

5.3.6.4 The default cooking time value at power up should be 00:00.

Rationale: Assume the operator would want an initial cooking time value of 00:00.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.4 of Requirements Specification>

5.3.6.5 While the Timed Cooking function is active, the current cooking time value should be displayed on the LCD screen.

Rationale: The operator should be able to clearly see the remaining cooking time.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.5 of Requirements Specification>

5.3.6.6 If the cooking time value is greater than zero and the Start keypad button is pressed:

a) the oven shall be turned on and

b) the cooking time value shall be decremented by one second each second until either the Stop/Clear button is pressed or until the cooking time value reaches 0:00.

<See Section 1.2.2 of the Concept Document>

Rationale: The display has a resolution of one second so it makes sense to decrement the timer value every second.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.6 of Requirements Specification>

5.3.6.7 If the Stop/Clear button on the microwave keypad is pressed while the oven is on:

a) the oven shall be shut off,

b) the cooking time value shall stop decrementing, and

c) the cooking time value at the time the Stop/Clear button was pushed shall be displayed on the LCD display.

<See Section 1.2.2 of Concept Document>

Rationale: Pressing the Stop/Clear button one time during cooking should stop the cooking.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.7 of Requirements Specification>

5.3.6.8 If the microwave door is opened during cooking:

a) **Within 500 msec the oven shall shut off**

   Rationale: To prevent a safety hazard the microwave must immediately shut off the oven if the door is opened during cooking.

b) **a buzzer shall sound .**

   Rationale: The buzzer signifies that the oven has shut off.

c) **the cooking time value shall stop decrementing**

   Rationale: The operator may wish to check the temperature of the item being cooked then resume cooking if the item is not yet hot enough

d) **the cooking time value when the door was opened shall be displayed on the LCD display.**

<See Section 1.2.2 of Concept Document>

Rationale: The operator should be able to see the amount of time left to cook on the display

Non Functional Requirement: The buzzer shall sound for three seconds +/- 200 msec.

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.8 of Requirements Specification>

5.3.6.9 If cooking was stopped before the time reached 00:00, either from the door being opened or from the Stop/Clear key being pressed, and then the Start keypad is pressed:

a) **the oven shall be turned back on and**
b) **the remaining time to cook value shall once again begin decrementing.**

Rationale: If the cooking was paused pressing the Start button should restart the cooking and the decrementing of the cooking time value.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.9 of Requirements Specification>

5.3.6.10 If cooking was stopped before the time reached 00:00, and the Stop/Clear button is pushed a second time:

a) the cooking time value shall immediately be set to 00:00 and

b) the buzzer shall sound to signify that the cooking has stopped.

<See Section 1.2.2 of the Concept Document>

Rationale: It is assumed that if the operator pushes the Stop/Clear button a second time then the operator wants to exit the timed cooking function.

Non Functional Requirement: The buzzer shall sound for 2 seconds, +/- 200 msec

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.10 of Requirements Specification>

5.3.6.11 Within 200 msec after the timer decrements from the value of 00:01 to the value 00:00,

a) the oven shall be shut off and

b) the buzzer shall sound.

<See Section 1.1.6 of the Concept Document>

Rationale: The buzzer sound will notify the operator that the cooking time value has reached 00:00.

Non Functional Requirement: The buzzer shall sound for three seconds +/- 200 msec.

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.11 of Requirements Specification>

5.3.6.12 When the oven is on, the operator shall not be allowed to change the power level.

Rationale: For safety reasons the power level will not be allowed to change while the oven is on.

Non Functional Requirement: None

Inspection results:

Checks passed: 1) Unique number? Yes
2) Amalgamation? No
3) Clear? Yes
4) Verifiable? Yes

Checks failed: None

<See Table 6.6.12 of Requirements Specification>

6 REQUIREMENT SPECIFICATION (FORM BASED)

6.1 GENERAL REQUIREMENTS

6.1.1 Table

5.3.1.2 <See Requirement 5.3.1.1 **Each time a keypad button is pressed a tone shall sound**>

Function:	Sound tone when button pushed
Description:	A tone should sound each time the operator pushes a button on the keypad.
Inputs:	Signal from the keypad that a button was pressed
Outputs:	A tone to the sound output device
Destination:	Sound output device
Pre-condition:	A keypad button must be pressed
Post-condition:	A tone will be heard
Side effects:	None

<Forward Pointer – put pointer to design that implements this, tests that test this or subrequirements – point to any place that uses this requirement in the life cycle documents >

6.1.2 Table

<See Requirement 5.3.1.2 – must copy the requirement here >

Function:	Sound tone when door is opened or closed
Description:	A tone shall sound whenever the microwave door is opened or closed.
Inputs:	Signal from door sensor
Outputs:	A tone signal to sound output device
Destination:	Sound output device
Pre-condition:	The microwave door must be opened or closed
Post-condition:	None
Side effects:	None

<Forward Pointer>

6.1.3 Table

<See Requirement 5.3.1.3 –must copy the actual requirement here for all requirements specified>

Function:	Default to Clock Function
Description:	When the microwave is powered up it should default to the Clock function.
Inputs:	None
Outputs:	The default time should be sent to the LCD screen

Destination:	LCD screen
Pre-condition:	The microwave is just powered on
Post-condition:	The default clock time should appear on the LCD screen
Side effects:	None

<Forward Pointer>

6.1.4 Table

<See Requirement 5.3.1.4 – must copy requirement here - do this for rest of requirements >

Function:	Microwave functions supported
Description:	The microwave shall provide a timer, a clock, six auto cook presets, an auto defrost mode, and a timed cooking mode.
Inputs:	Buttons on keypad
Outputs:	The desired function is selected depending on which buttons the operator presses
Destination:	Mode control portion of microwave software
Pre-condition:	A button on keypad is pressed to choose function (e.g., the Clock button selects the Clock function).
Post-condition:	The desired microwave function is selected and displayed on the LCD screen (see requirement 5.3.1.5).
Side effects:	None

<Forward Pointer>

6.1.5 Table

<See Requirement 5.3.1.5>

Function:	Display active microwave function
Description:	The current microwave function should be displayed on the LCD screen.
Inputs:	The active function indicator
Outputs:	The active function indicator
Destination:	LCD screen.
Pre-condition:	One of the microwave functions, or the default function, is active
Post-condition:	The active function is identified on the LSC screen
Side effects:	None

<Forward Pointer>

6.2 CLOCK

6.2.1 Table

<See Requirement 5.3.2.1>

Function:	Activate Clock
Description:	The user pressed the Clock pad on keypad
Inputs:	The user inputs the current time from the keypad.
Outputs:	The desired time
Destination:	LCD screen
Pre-condition:	Clock keypad has to be pressed
Post-condition:	The Clock function should be active as indicated on the LCD screen.
Side effects:	None

<Forward Pointer>

6.2.2 Table

<See Requirement 5.3.2.2>

Function:	Default clock value
Description:	The Clock time's default is at twelve (12:00).
Inputs:	None
Outputs:	The desired time.
Destination:	LCD screen.
Pre-condition:	The microwave was just powered on
Post-condition:	The clock is set to the default value and this value is displayed on the LCD screen.
Side effects:	None

<Forward Pointer>

6.2.3 Table

<See Requirement 5.3.2.3>

Function:	Clock settable in one-minute increments
Description:	The clock should be settable to any minute of the day.
Inputs:	The user inputs the current time from the keypad.
Outputs:	The desired time
Destination:	LCD screen
Pre-condition:	None
Post-condition:	The desired time is displayed on the LCD screen.
Side effects:	None

<Forward Pointer>

6.2.4 Table

<See Requirement 5.3.2.4>

Function:	Displaying the current time
Description:	The current time is displayed on the LCD screen.
Inputs:	None
Outputs:	The current time
Destination:	LCD screen
Pre-condition:	The clock function is active.
Post-condition:	The current time is displayed on LCD screen.
Side effects:	None

<Forward Pointer>

6.2.5 Table

<See Requirement 5.3.2.5>

Function:	Setting the clock
Description:	Pads 0 through 9 are used to set the clock.
Inputs:	The user inputs the current time from the keypad.
Outputs:	The current time
Destination:	LCD screen
Pre-condition:	The clock function is active.
Post-condition:	The LCD clock value displays the time that was set by the user.
Side effects:	None

<Forward Pointer>

6.2.6 Table

<See Requirement 5.3.2.6>

Function:	Starting the clock
Description:	Once the clock has been started, the current time value should increment each minute..
Inputs:	The Start pad is pressed
Outputs:	The current time
Destination:	LCD screen
Pre-condition:	The desired time has been entered.
Post-condition:	Time is incrementing on the LCD screen.
Side effects:	None

<Forward Pointer>

6.2.7 Table

<See Requirement 5.3.2.7>

Function:	The colon symbol
Description:	The colon symbol separates minutes from hours.
Inputs:	None
Outputs:	The colon symbol
Destination:	LCD screen
Pre-condition:	The clock function is active.
Post-condition:	The colon separates the hours from the minutes on the LCD screen.
Side effects:	None

<Forward Pointer>

6.2.8 Table

<See Requirement 5.3.2.8>

Function:	The flashing colon symbol
Description:	The colon is flashed when the clock is running.
Inputs:	None
Outputs:	Signal to turn the colon symbol on and off
Destination:	LCD screen
Pre-condition:	The clock function is active.
Post-condition:	The flashing colon indicates that the clock is running.
Side effects:	None

<Forward Pointer>

6.2.9 Table

<See Requirement 5.3.2.9>

Function:	The non-flashing colon symbol
Description:	The colon is not flashed while time is being set.
Inputs:	None
Outputs:	Signal to keep colon on.
Destination:	LCD screen.
Pre-condition:	The clock function is active
Post-condition:	The colon remaining on indicating that the clock is not running.
Side effects:	None

<Forward Pointer>

6.3 TIMER

6.3.1 Table

<See Requirement 5.3.3.1>

Function:	Enter Timer function
Description:	Enter Timer function in microwave software
Inputs:	Timer pad on the keypad
Outputs:	Signal to turn on Timer function light
Destination:	LCD screen
Pre-condition:	Any function could be active
Post-condition:	Timer function selected as indicated on display
Side effects:	None

<Forward Pointer>

6.3.2 Table

<See Requirement 5.3.3.2>

Function:	Default timer value
Description:	Set timer value to default value (00:00)
Inputs:	Initial power applied to microwave
Outputs:	Timer value initialized to zero
Destination:	Timer function
Pre-condition:	Timer value not initialized
Post-condition:	Timer value initialized to zero
Side effects:	None

<Forward Pointer>

6.3.3 Table

<See Requirement 5.3.3.3>

Function:	Set timer value.
Description:	The operator sets the desired timer value
Inputs:	Pads 0 through 9 on keypad
Outputs:	Timer value to be used by Timer function
Destination:	Timer function
Pre-condition:	Timer value unknown
Post-condition:	Timer value set to value entered by user.
Side effects:	None

<Forward Pointer>

6.3.4 Table

<See Requirement 5.3.3.4>

Function:	Decrement timer value
Description:	Decrement the timer value once per second.
Inputs:	Original Timer value

Outputs:	New Timer value
Destination:	Timer function
Pre-condition:	Timer value
Post-condition:	Timer value minus one
Side effects:	None

<Forward Pointer>

6.3.5 Table

<See Requirement 5.3.3.5>

Function:	Sound buzzer at time zero
Description:	When the timer value decrements to zero sound the buzzer.
Inputs:	Timer value equal to zero
Outputs:	Signal to sound buzzer
Destination:	Microwave buzzer
Pre-condition:	Timer value decrements to zero
Post-condition:	Buzzer sounds
Side effects:	None

<Forward Pointer>

6.3.6 Table

<See Requirement 5.3.3.6>

Function:	Display Current Timer value
Description:	Display the current timer value on the LCD display.
Inputs:	Timer value
Outputs:	Timer value
Destination:	LCD display
Pre-condition:	N/A
Post-condition:	Current Timer value shown on LCD display
Side effects:	None

<Forward Pointer>

6.3.7 Table

<See Requirement 5.3.3.7>

Function:	Stop Timer
Description:	Stop Timer when the Stop/Clear key is pressed once.
Inputs:	Stop/Clear key on keypad
Outputs:	Stop signal set
Destination:	Timer function.
Pre-condition:	Timer active and counting down time.
Post-condition:	Timer stopped.
Side effects:	None

<Forward Pointer>

6.3.8 Table

<See Requirement 5.3.3.8>

Function:	Restart Timer
Description:	Restart stopped timer when the Start key is pressed.
Inputs:	Start key on keypad
Outputs:	Start input set
Destination:	Timer function
Pre-condition:	Timer stopped because Stop/Clear key was pressed
Post-condition:	Timer begins decrementing again
Side effects:	None

<Forward Pointer>

6.3.9 Table

<See Requirement 5.3.3.9>

Function:	Cancel Timer
Description:	Cancel timer if the Stop/Clear key is pressed a second time.
Inputs:	Stop/Clear key on keypad
Outputs:	Cancel input set
Destination:	Timer function
Pre-condition:	Timer stopped because Stop/Clear key was pressed.
Post-condition:	Timer stopped and Timer value reset to zero.
Side effects:	None

<Forward Pointer>

6.3.10 Table

<See Requirement 5.3.3.10>

Function:	Continue timer countdown if door opens.
Description:	Do not stop the timer countdown if the microwave door is opened during timer operation.
Inputs:	Sensor on microwave door
Outputs:	Stop input is not set
Destination:	Timer function
Pre-condition:	Timer value is being decremented each second
Post-condition:	Timer continues counting down
Side effects:	None

<Forward Pointer>

6.4 AUTO COOK PRESETS

6.4.1 Table

<See Requirement 5.3.4.1>

Function:	Automatically heat common food items
Description:	Six preset functions: Popcorn, Potato, Beverage, Pizza, Frozen Entrée, and Fresh Vegetables
Inputs:	The user presses one of the preset keys
Outputs:	Time and power level are set by control software
Destination:	Time and power level are displayed on LCD screen
Pre-condition:	Preset has been selected
Post-condition:	Food item has been heated at the proper power level for the correct amount of time.
Side effects:	None

<Forward Pointer>

6.4.2 Table

<See Requirement 5.3.4.2>

Function:	Unique cooking times and power level for each auto cook preset button
Description:	The capability should be provided for each auto cook preset to have its own predefined cooking time and power level settings.
Inputs:	Auto cook preset buttons on keypad
Outputs:	Predefined cooking time and power level
Destination:	Control section of microwave software
Pre-condition:	An auto cook preset button is pressed to activate auto cook preset function
Post-condition:	The predefined cooking time and power level associated with the auto cook preset are selected.
Side effects:	None

<Forward Pointer>

6.4.3 Table

<See Requirement 5.3.4.3>

Function:	Begin auto cook
Description:	If auto-cook preset button has been pressed (cooking time is greater than zero), begin timed cooking when start button is pressed.
Inputs:	Start button on keypad
Outputs:	Auto cook preset mode selection and cooking time
Destination:	LCD screen
Pre-condition:	An auto cook preset key is pressed followed by the Start button being pushed
Post-condition:	The Auto Cook Preset function is active as shown on the LCD screen with the correct time to cook value decrementing on the screen.
Side effects:	None

<Forward Pointer>

6.5 AUTO DEFROST

6.5.1 Table

<See Requirement 5.3.5.1>

Function:	Automatically sets defrost time and power level for frozen meat items depending on the weight of the item entered by the user.
Description:	One function labeled Auto Defrost located on bottom row of numerical keypad
Inputs:	Type of meat, weight of meat.
Outputs:	Prompts for type of meat and weight of meat
Destination:	Prompts appear on LCD screen
Pre-condition:	Auto Defrost has been selected
Post-condition:	Meat item has been defrosted for the correct amount of time at the correct power level
Side effects:	None

<Forward Pointer>

6.5.2 Table

<See Requirement 5.3.5.2>

Function:	Type of meat to be defrosted
Description:	Offers a choice of Beef, Poultry, or Fish
Inputs:	User presses a number indicated which type of meat to defrost.
Outputs:	Control software sets time and power level
Destination:	Time and power level displayed on LCD
Pre-condition:	Auto Defrost has been selected
Post-condition:	The time and power level are automatically set
Side effects:	None

<Forward Pointer>

6.5.3 Table

<See Requirement 5.3.5.3>

Function:	Menu for meat selection
Description:	Displays a menu listing the choices of Beef, Poultry, and Fish
Inputs:	User will press a numeric key on the keypad to select a choice from the menu
Outputs:	Choice and power level are set and displayed
Destination:	Message containing choice and power level are sent to LCD screen

Pre-condition:	Auto Defrost has been selected
Post-condition:	Time and power levels are automatically set. Message indicating selection and power level are displayed.
Side effects:	None

<Forward Pointer>

6.5.4 Table

<See Requirement 5.3.5.4>

Function:	Set the weight of the meat item
Description:	The user must enter the approximate weight of the meat item.
Inputs:	Numerical values entered on the numeric keypad.
Outputs:	Weight will be displayed in pounds and ounces.
Destination:	Weight will be displayed on the LCD
Pre-condition:	Type of meat has been selected
Post-condition:	Defrosting time is set in accordance with the weight of the meat item.
Side effects:	None

<Forward Pointer>

6.5.5 Table

<See Requirement 5.3.5.5>

Function:	Prompt for weight of meat item
Description:	Control software will prompt user for approximate weight of meat.
Inputs:	Weight will be entered in pounds and ounces on numeric keypad.
Outputs:	Weight will be set by control software and displayed in LCD screen.
Destination:	Control software and LCD screen
Pre-condition:	Type of meat has been selected
Post-condition:	Weight of meat will be set. Defrosting time will be set accordingly.
Side effects:	None

<Forward Pointer>

6.5.6 Table

<See Requirement 5.3.5.6>

Function:	Start key will begin defrosting process
Description:	User must press the Start key to begin defrosting after all information has been entered.
Inputs:	User presses the Start key
Outputs:	Control software begins defrosting process
Destination:	Countdown of defrosting time appears on LCD screen
Pre-condition:	Type of meat has been selected and weight has been entered
Post-condition:	Meat is defrosted at the proper power level and for the correct amount of time.
Side effects:	None

<Forward Pointer>

6.6 TIMED COOKING

6.6.1 Table

<See Requirement 5.3.6.1>

Function:	Enter power level.
Description:	Operator enters the desired power level to be used for timed cooking.
Inputs:	Power Level pad followed by a pad 1 through 9 on keypad
Outputs:	Desired power level.
Destination:	Timed cooking function
Pre-condition:	None
Post-condition:	Power level set to desired value
Side effects:	None

<Forward Pointer>

6.6.2 Table

<See Requirement 5.3.6.2>

Function:	Default power level
Description:	The default power level should be Level 9.
Inputs:	None
Outputs:	Default power level set equal to 9
Destination:	Timed cooking function.
Pre-condition:	None
Post-condition:	Default power level set to 9

Side effects:	None

<Forward Pointer>

6.6.3 Table

<See Requirement 5.3.6.3>

Function:	Enter cooking time
Description:	Operator enters the desired cooking time.
Inputs:	Pads 0 through 9 on keypad
Outputs:	Desired cooking time
Destination:	Timed cooking function
Pre-condition:	None
Post-condition:	Cooking time set to desired value
Side effects:	None

<Forward Pointer>

6.6.4 Table

<See Requirement 5.3.6.4>

Function:	Default cooking time value
Description:	The default cooking time should be 00:00.
Inputs:	None
Outputs:	Default cooking time of 00:00
Destination:	Timed cooking function
Pre-condition:	None
Post-condition:	Cooking time set to default value
Side effects:	None

<Forward Pointer>

6.6.5 Table

<See Requirement 5.3.6.5>

Function:	Display cooking time
Description:	Current cooking time is displayed on the LCD screen
Inputs:	Current cooking time
Outputs:	Current cooking time
Destination:	LCD screen
Pre-condition:	None
Post-condition:	Current cooking time is displayed on the LCD screen
Side effects:	None

<Forward Pointer>

6.6.6 Table

<See Requirement 5.3.6.6>

Function:	Begin timed cooking
Description:	If cooking time is greater than zero begin timed cooking when Start pad is pressed.
Inputs:	Start pad
Outputs:	Current cooking time value
Destination:	Timed cooking function
Pre-condition:	A cooking time value greater than zero
Post-condition:	Oven is turned on and cooking time value is decrementing
Side effects:	None

<Forward Pointer>

6.6.7 Table

<See Requirement 5.3.6.7>

Function:	Stop timed cooking
Description:	If Stop/Clear pad is pressed while oven is on the cooking should stop and the current cooking time value should be displayed.
Inputs:	Stop/Clear pad
Outputs:	Stop signal
Destination:	Timed cooking function
Pre-condition:	Oven on
Post-condition:	Oven shut off and cooking time stops decrementing (with current cooking time value displayed on LCD screen)
Side effects:	None

<Forward Pointer>

6.6.8 Table

<See Requirement 5.3.6.8>

Function:	Door opened during cooking.
Description:	If door is opened during cooking shut oven off, sound buzzer, and stop decrementing the cooking time value.
Inputs:	Signal from door sensor indicating the door has been opened
Outputs:	Stop signal set
Destination:	Timed cooking function.
Pre-condition:	None
Post-condition:	Oven shut off, buzzer sounds, cooking time stops decrementing (with current cooking time value displayed on LCD screen)
Side effects:	None

<Forward Pointer>

6.6.9 Table

<See Requirement 5.3.6.9>

Function:	Restart cooking.
Description:	Operator presses Start pad to restart timed cooking that has been halted.
Inputs:	Start pad
Outputs:	Start signal
Destination:	Timed cooking function.
Pre-condition:	None
Post-condition:	Oven is turned on and cooking time begins decrementing
Side effects:	None

<Forward Pointer>

6.6.10 Table

<See Requirement 5.3.6.10>

Function:	Cancel timed cooking
Description:	Operator presses Stop/Clear pad after cooking has already been halted to cancel timed cooking.
Inputs:	Stop/Clear pad
Outputs:	Cancel signal set
Destination:	Timed cooking function
Pre-condition:	Timed cooking halted
Post-condition:	Power level set to desired value
Side effects:	None

<Forward Pointer>

6.6.11 Table

<See Requirement 5.3.6.11>

Function:	Sound buzzer at time zero
Description:	When the cooking time value decrements to zero sound the buzzer.
Inputs:	Cooking time value equal to zero
Outputs:	Signal to sound buzzer
Destination:	Microwave buzzer
Pre-condition:	Cooking time value decrements to zero
Post-condition:	Buzzer sounds
Side effects:	None

<Forward Pointer>

6.6.12 Table

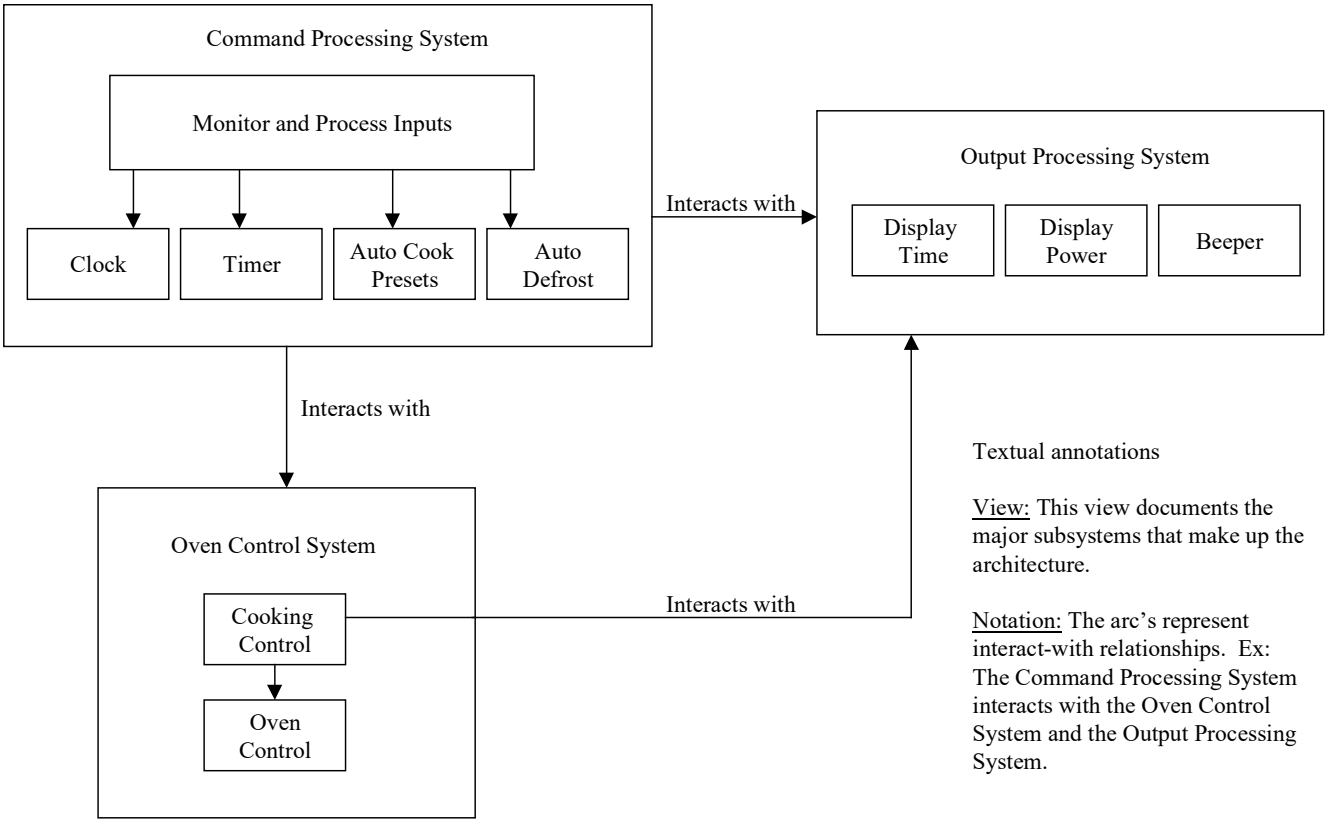
<See Requirement 5.3.6.12>

Function:	Changing power level while cooking
Description:	For safety reasons the operator should not be allowed to change the power level while the oven is turned on.
Inputs:	Power level key followed by a keypad entry of 1 through 9
Outputs:	Unchanged power level
Destination:	Timed cooking function
Pre-condition:	None
Post-condition:	No change in the power level
Side effects:	None

<Forward Pointer>

7 ARCHITECTURAL DESIGN

7.1 STRUCTURAL VIEW



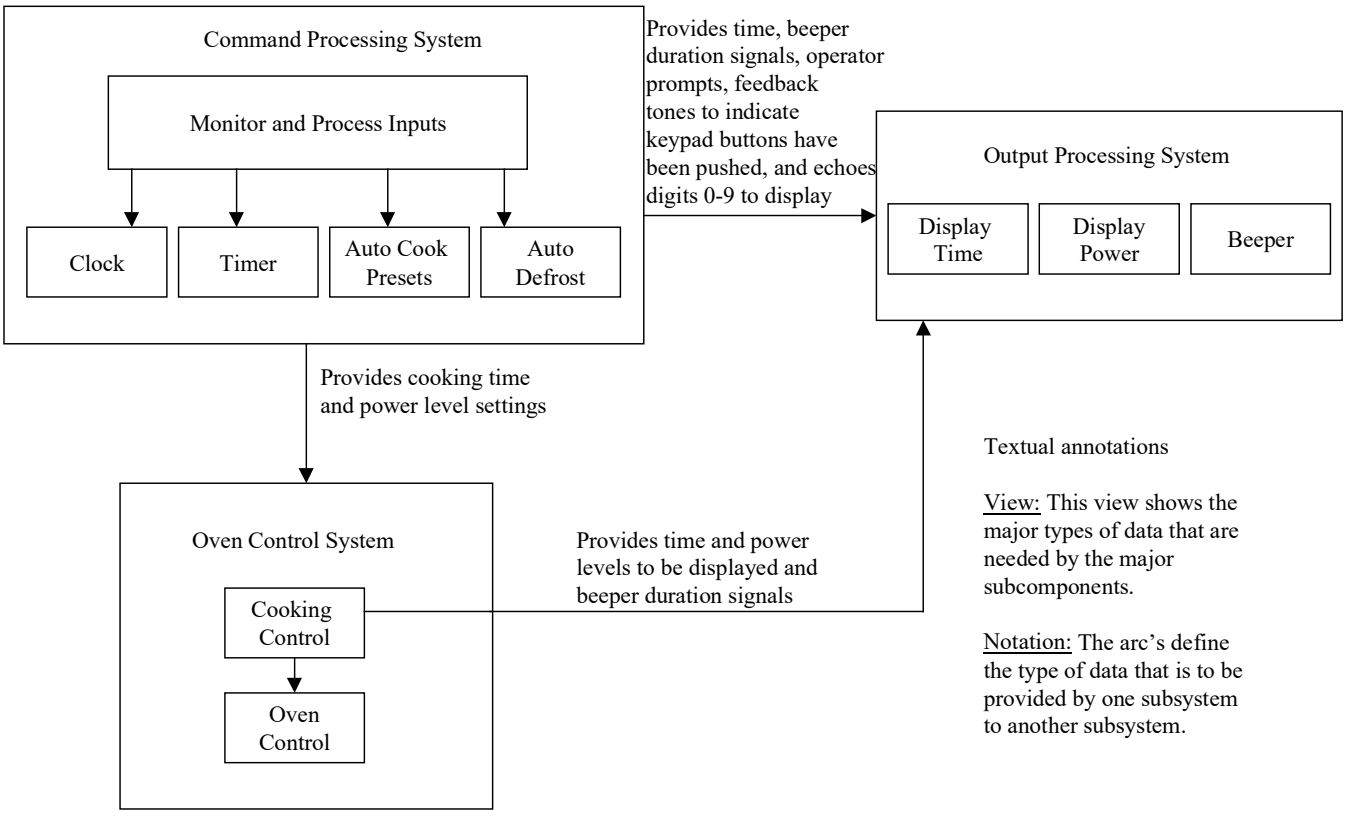
Structural View of Architecture: Microwave Oven

Data Dictionary:

Term	Description	Type
Auto Cook Presets	Processes input data containing manufacturer-defined presets for heating common food items.	Sub-system
Auto Defrost	Processes input data containing power level and time for defrosting frozen food item.	Sub-system
Beeper	Processes output signals to the beeper device.	Sub-system
Clock	Processes input data containing clock-setting parameters.	Sub-system
Command Processing	Processes all inputs from the user.	System
Cooking Control	Processes output data from Command Processing System and sends command signals to the Oven Control sub-system.	Sub-system

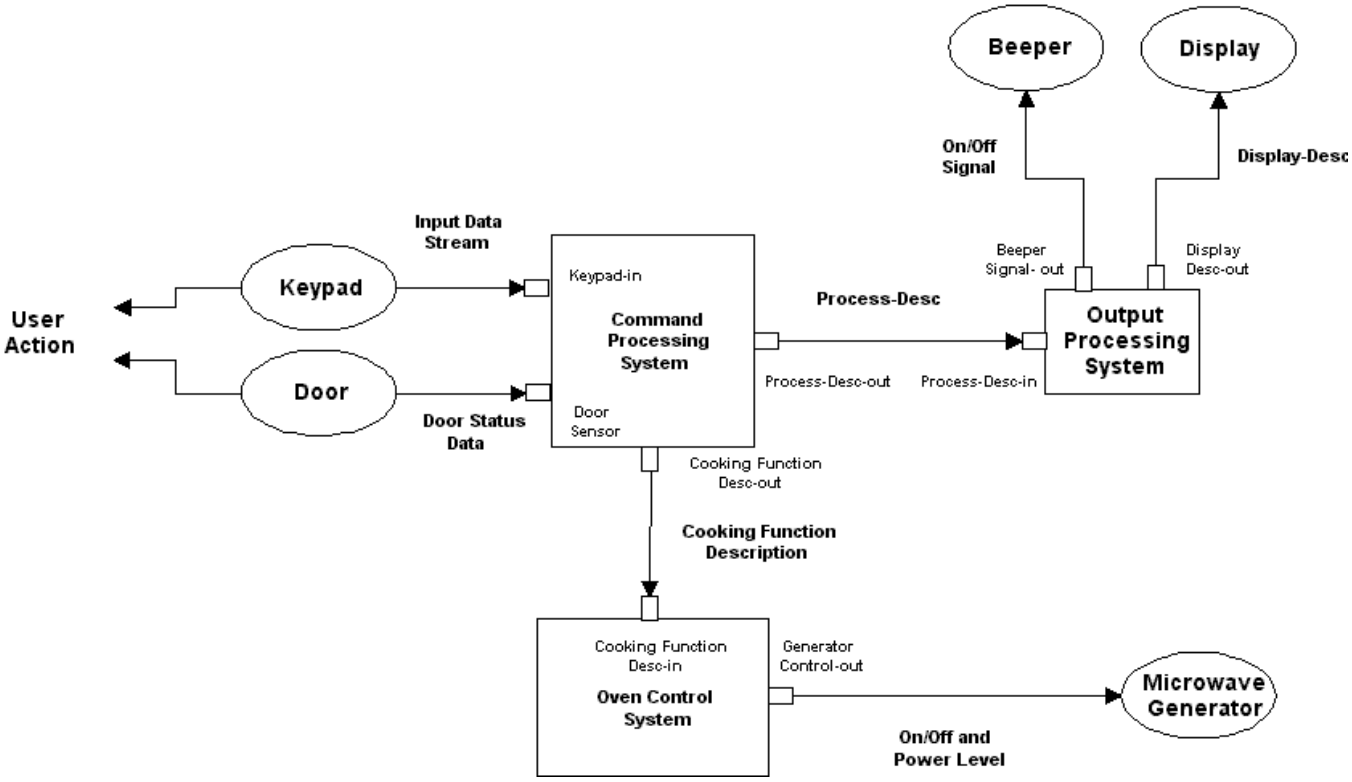
Display Power	Processes output data containing the current power level that is encoded for display on the LCD.	Sub-system
Display Time	Processes output data containing the current time of day value that is encoded for display on the LCD.	Sub-system
Monitor / Process Inputs	Processes pre-processed inputs and controls system state based on those inputs.	Sub-system
Output Processing	Processes output data and converts it to signals and codes that can be used by the target output devices.	System
Oven Control (sub-system)	Processes data sent from the Cooking Control sub-system and converts the data into the correct format for the cooker component.	Sub-system
Oven Control (system)	Processes output data and converts it to signals that can be used by the cooker component.	System

7.2 DATA VIEW



Data View of Architecture: Microwave Oven

7.3 GRAPHICAL COMPONENT INTERCONNECTION/INTERFACE VIEW



Graphical Component Interconnection/Interface View

Data Dictionary:

Term	Description	Type
Beeper	Beeper component that encapsulates beeper functionality.	Component
Beeper Signal-out	Output from the Output Processing System component that contains signals to the beeper device.	Output Data
Command Processing System	Controller component that encapsulates controller functionality.	Component
Cooking Function Description	Cooking function description data that flows from the Command Processing System component to the Oven Control System component. This data identifies the cooking function selected.	Data Flow
Cooking Function Desc-in	Oven Control System input cooking data.	Input Data
Cooking Function Desc-out	Command Processing System component output cooking data.	Output Data
Display	Display component that encapsulates display functionality.	Component

Display Desc	Data that flows from the Output Processing System component to the Display component that contains signals to the display device.	Data Flow
Display Desc-out	Output from the Output Processing System component that contains signals to the display device.	Output Data
Door	Door component that encapsulates door functionality.	Component
Door Sensor	Input data to theCommand Processing System (Controller) component that contains door status data.	Input Data
Door Status Data	Data that flows from the door latch sensor to the Command Processing System (Controller) component.	Data Flow
Generator Control-out	Oven Control System component output that contains the control codes for the generator.	Output Data
Input Data Stream	Data that flows from the Keypad component to the Command Processing System (Controller) component. This data contains keycode data from the keypad.	Data Flow
Keypad	Keypad component that encapsulates keypad functionality.	Component
Keypad-in	Input data to the Command Processing System (Controller) component that contains keycode data from the keypad.	Input Data
Microwave Generator	Microwave Generator (cavity magnetron) component that encapsulates generator functionality.	Component
On/Off and Power Level	Data that flows from the Oven Control System component to the Microwave Generator component. This data that contains the current status and power level of the generator.	Data Flow
On / Off Signal	Data that flows from the Output Processing System component to the Beeper component. This data contains the status of the beeper device.	Data Flow
Output Processing System	Output Processing System component that encapsulates output processing system functionality.	Component
Oven Control System	Oven Control System component that encapsulates oven control functionality.	Component
Process-Desc	Process-related data that flows from the Command Processing System component to the Output Processing System component.	Data Flow
Process-Desc-In	Input data to the Output Processing System component that contains cooking process-related data.	Input Data
Process-Desc-Out	Output from the Command Processing System component that contains cooking process-related data.	Output Data

7.4 NOTATION FOR INTERCONNECTION/INTERFACE VIEW

Component: Command Processing System
Component Type: Hardware/Software Control Component

Input Port List:
Port name: Keypad-in **Data type:** Keypad Commands;
Port name: Door Status-in **Data type:** Sensor;

Output Port List:

Port name: Cooking-Function-Desc-Out **Data type:** Cooking-Function-Desc;
Port name: Process-Desc-Out **Data type:** Process-Desc;

Direction Flow:

Cooking-Function-Desc-Out => Oven-Control-System.Cooking-Func-Desc-in;
Process-Desc-Out => Output-Processing-System.Process-Desc-in;

General Behavior Spec:

The input data is received from the keypad and the door sensor. The command processing system sorts the input and directs it to the appropriate sub-system for further processing.

Contraints:

The Command Processing System cannot output data directly to hardware (i.e, Display or Microwave Generator). All signals to hardware must be received from the subsystem governing their operation.

Component: Oven Control System
Component Type: Hardware Control Component

Input Port List:

Port name: Cooking-Function-Desc-In **Data type:** Cooking Function Description;

Output Port List:

Port name: Generator-Control-Out **Data type:** On/Off OR Power Level Signals;

Direction Flow:

Generator-Control-Out => Microwave Generator;

General Behavior Spec:

The Oven Control System receives data from the Command Processing System. This data is used to activate or deactivate as well as set the power level of the microwave generator.

Constraints:

The Oven Control System can receive data only from the Command Processing System of a Cooking-Function-Description type.

Component: Output Processing System
Component Type: Hardware Control Component

Input Port List:

Port name: Process-Desc-in **Data type:** Process-desc;

Output Port List:

Port name: Beeper-Signal-out **Data type:** On/Off signal;
Port name: Display-Desc-out **Data type:** Display-Desc;

Direction Flow:

Beeper-Signal-out => Beeper;
Display-Desc-out => Display;

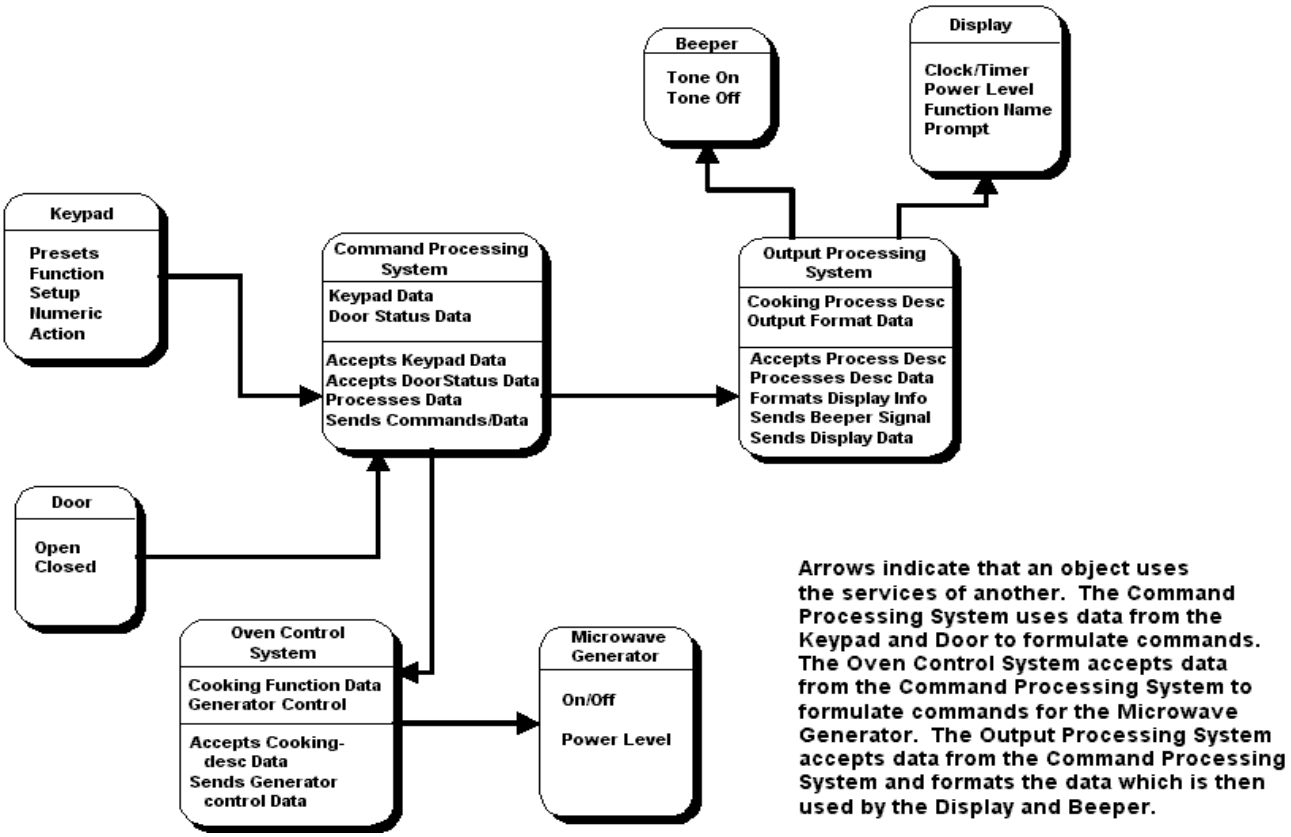
General Behavior Spec:

The Output Processing System analyzes process descriptions sent to it by the Command Processing System. It then determines the appropriate format and destination (Beeper or Display) for an output signal.

Constraints:

The Output Processing System can only accept data from the Command Processing System. The data must be of type Process-Desc.

7.5 OBJECT ORIENTED DECOMPOSITION VIEW OF ARCHITECTURE



Object Oriented Decomposition View

Data Dictionary:

Term	Description	Type
Accepts Cooking-desc Data	Oven Control System operation that accepts cooking data.	Method
Accepts DoorStatus Data	Command Processing System (Controller) operation that accepts data from the door sensor.	Method
Accepts Keypad Data	Command Processing System (Controller) operation that accepts keycode data from the keypad.	Method
Accepts Process Desc	Output Processing System operation that accepts process-related data.	Method
Action	Keypad action key.	Attribute
Beeper	Beeper entity that encapsulates beeper attributes.	Class

Clock / Timer	Clock and Timer attributes of the Display entity.	Attribute
Closed	Status attribute of the Door class.	Attribute
Command Processing System	Controller entity that encapsulates controller object attributes and operations.	Class
Cooking Function Data	Oven Control System attribute that identifies the cooking function selected.	Attribute
Cooking Process Desc	Output Processing System attribute that contains cooking process-related data.	Attribute
Display	Display entity that encapsulates display object attributes.	Class
Door	Door entity that encapsulates door object attributes.	Class
Door Status Data	Command Processing System (Controller) attribute that contains door status data.	Attribute
Formats Display Info	Output Processing System operation that formats data that will be sent to the display device.	Method
Function	Keypad Function key.	Attribute
Function Name	Attribute of the Display entity that contains the name of the selected function.	Attribute
Generator Control	Oven Control System attribute that contains the control codes for the generator.	Attribute
Keypad	Keypad entity that encapsulates keypad object attributes.	Class
Keypad Data	Command Processing System (Controller) attribute that contains keycode data from the keypad.	Attribute
Microwave Generator	Microwave Generator (cavity magnetron) entity that encapsulates generator object attributes.	Class
Numeric	Keypad Numeric key.	Attribute
On/Off	Microwave Generator class attribute that contains the current status of the generator.	Attribute
Open	Status attribute of the Door class.	Attribute
Output Processing System	Output Processing System entity that encapsulates output processing system object attributes and operations.	Class
Power Level	Microwave Generator class attribute that contains the current power level of the generator.	Attribute
Power Level	Attribute of the Display entity that contains the power level selected by the user.	Attribute
Presets	Keypad Preset key.	Attribute
Processes Data	Command Processing System (Controller) operation that processes input data.	Method
Processes Desc Data	Output Processing System operation that processes descripton data.	Method
Prompt	Attribute of the Display entity that contains the prompt data to be displayed.	Attribute

Sends Beeper Signal	Output Processing System operation that sends signals to the beeper device.	Method
Sends Commands / Data	Command Processing System (Controller) operation that sends commands and data to the necessary components to maintain a valid system state.	Method
Sends Display Data	Output Processing System operation that sends signals to the display device.	Method
Sends Generator control Data	Oven Control System operation that sends the control data / signals to the generator.	Method
Tone Off	Attribute of the Beeper entity that contains the status of the beeper device.	Attribute
Tone On	Attribute of the Beeper entity that contains the status of the beeper device.	Attribute

8 FUNCTION-ORIENTED DESIGN

A function oriented design approach was chosen for the design phase of the microwave oven software development.

8.1 RATIONALE FOR CHOOSING FUNCTION-ORIENTED DESIGN

Function-oriented design seemed to fit the microwave oven software design better than an object-oriented design approach. The microwave oven software naturally broke up into a number of different functions (e.g., clock, timer, auto cook presets, auto defrost, etc.). How to implement this software design using an object-oriented approach was not as clear. The microwave oven does not have many objects with common attributes (such as the different types of employees used in an object-oriented example in the text). Also, the microwave oven does not appear to have any objects that could be defined to utilize the inheritance or multiple inheritance features that are available in object-oriented designs. Once a microwave oven is manufactured and sold, its software is never updated so maintenance was not as much of an issue for this design as it would be in some other software applications.

The members of our development group had more experience with function-oriented designs as compared to object-oriented designs which was another factor in choosing to do a functional design over object oriented design. It seemed best to stick to a straightforward function-oriented design given the short amount of time available to complete the design phase of this project.

Assuming that we kept the same software development team and continued to move on to other software development projects we would likely want to utilize an object-oriented approach, or a heterogeneous approach that incorporated both functional and object-oriented methodologies, in future designs. Object-oriented design does have definite benefits in certain applications (e.g., easier maintenance and easier reuse of existing code) but in this application a function-oriented design seemed to be the best approach to use.

8.2 MAJOR FUNCTIONS

The following major functions were identified for the microwave oven:

1. Monitor and Process Inputs Function – Decodes inputs from keypad and door sensor. Determines which microwave sub-function (Clock, Auto Defrost, etc.) is being activated and prompts the operator, if necessary, for additional information (e.g., type and weight of meat to defrost). Once all of the pertinent information is received this function calls the appropriate sub-function. This function also sends interrupts to the appropriate functions if the Stop/Clear button is pressed or if the oven door is opened during cooking.

Note: In a real-life software design process for the microwave oven software, the Monitor and Process Inputs Function should be broken down into further sub-functions. To limit the number of major functions to ten, for this assignment, this function was not broken down into smaller sub-functions.

2. Clock Function – Receives start time from Monitor and Process Inputs Function then increments the time value once per second and outputs the current time for display.

3. Timer Function – Receives initial timer value from Monitor and Process Inputs Function then decrements time once per second until zero seconds is reached. If a Stop interrupt signal is received during the count down, the time is halted until either a Stop/Clear signal is received (to cancel the timer countdown) or a Start signal is received (to continue the timer countdown). The current timer value is output once per second to be displayed and a signal is output to sound the beeper when the countdown reaches zero.

4. Auto Cook Presets Function – If an Auto Cook Preset button is pressed, this function will lookup the proper power level and cooking time for output to the Cooking Control Function.

5. Auto Defrost Function – Based on the type of meat and the weight of the meat being defrosted, this function determines the proper power level and cooking time values which are passed to the Cooking Control Function.

6. Cooking Control Function – Receives cooking time and power level inputs then sends signals to turn on the oven, at the proper power level, for the proper length of time. If a stop interrupt signal is received, indicating that the Stop/Clear button was pressed or the oven door was opened during cooking, the oven is shut off. Cooking is resumed if an input is received from the Monitor and Process Inputs Function indicating that the operator has pressed the Start button.

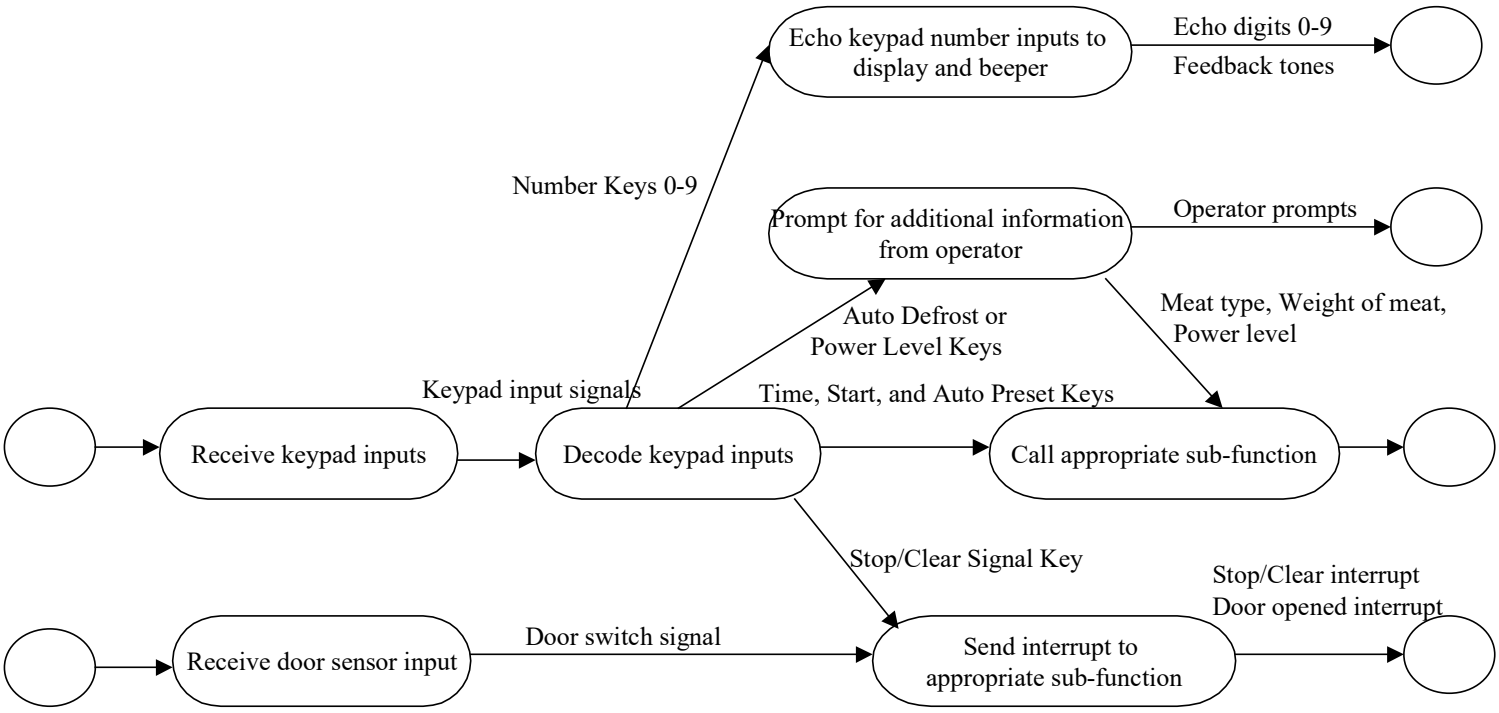
7. Oven Control Function – Sets the microwave generator to the proper power setting then turns the oven on and off as directed by the Cooking Control Function.

8. Display Time Function – Formats time properly then outputs time values to the LCD screen to for display.

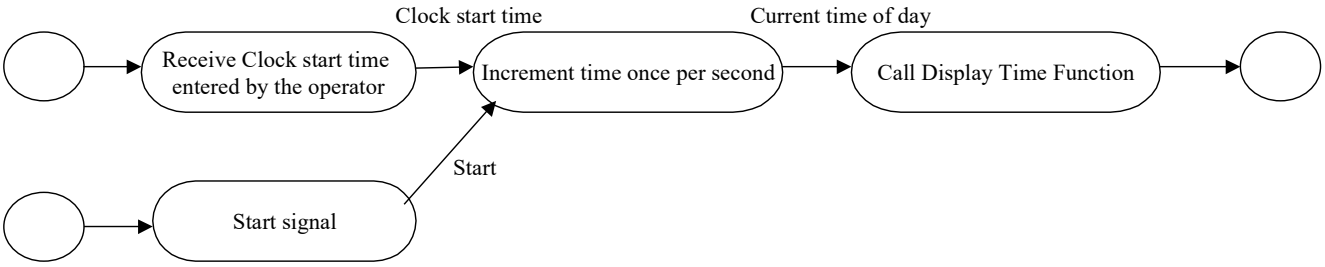
9. Display Power Function – Formats power level settings then outputs current power level settings to the LCD screen for display.

10. Beeper Function – Outputs signal to beeper hardware to sound a tone for desired length of time.

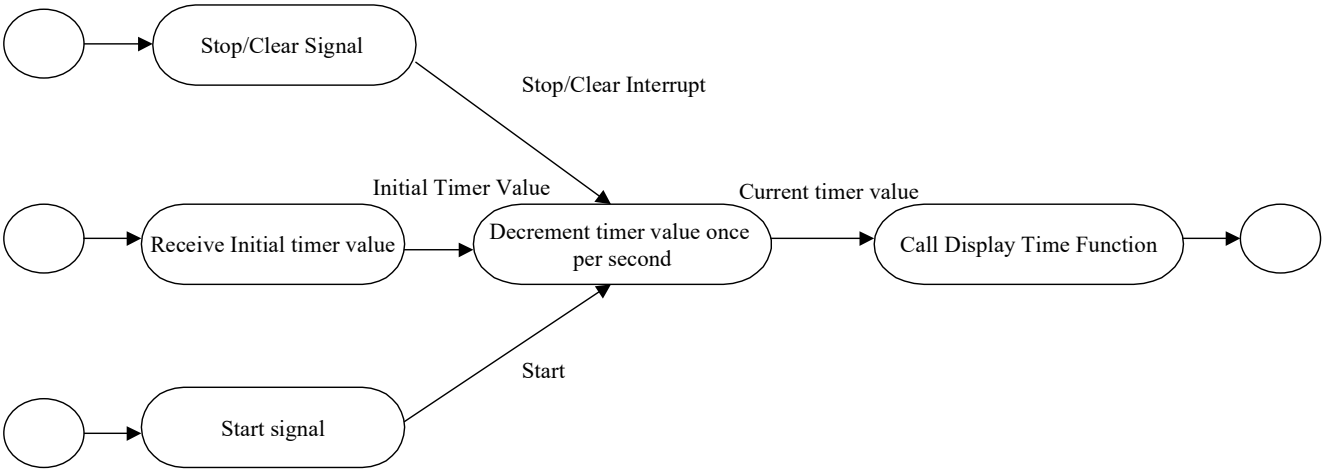
8.2.1 Data Flows for Major Functions



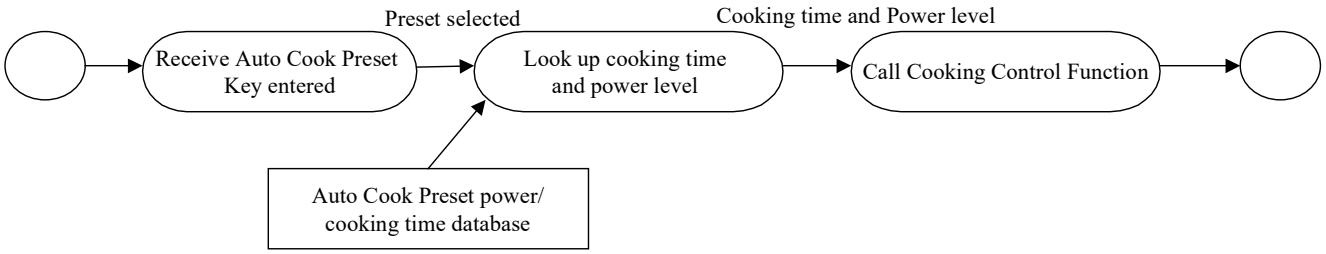
Data Flow for Monitor and Process Inputs Function



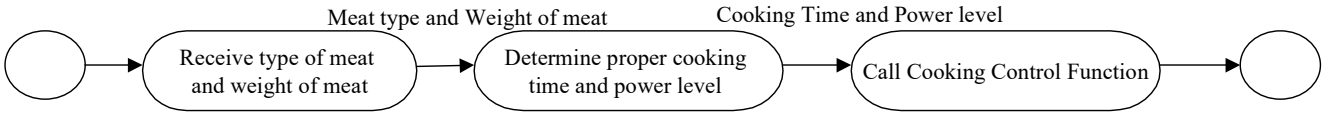
Data Flow for Clock Function



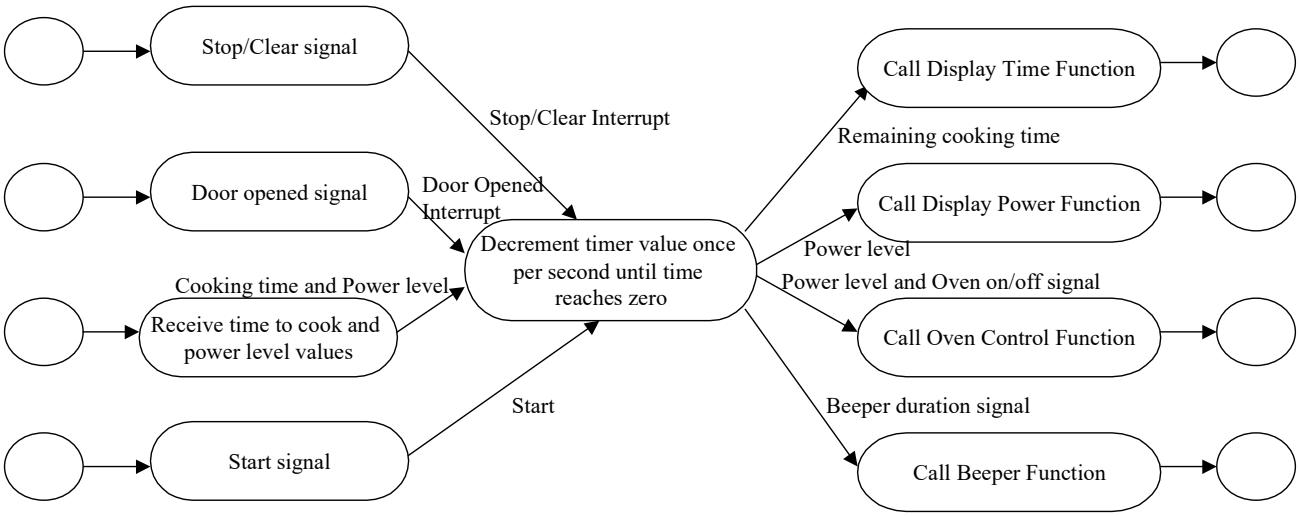
Data Flow for Timer Function



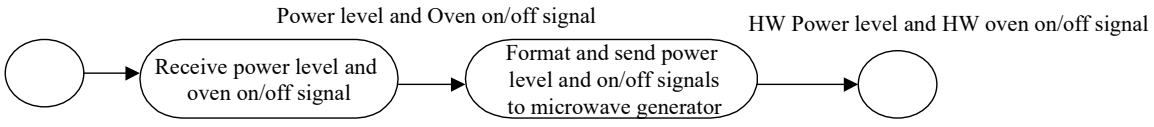
Data Flow for Auto Cook Presets Function



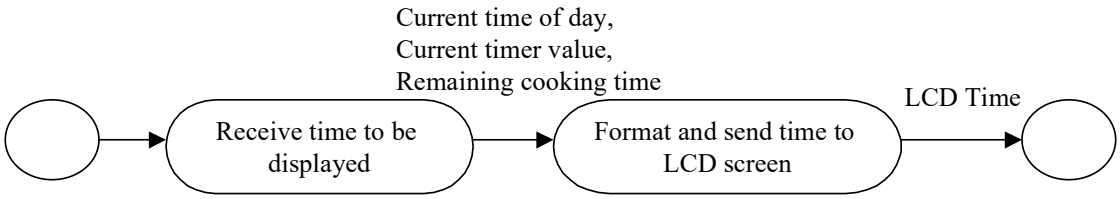
Data Flow for Auto Defrost Function



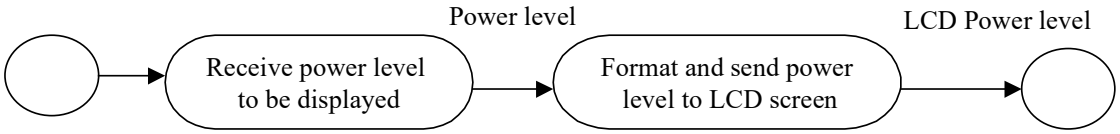
Data Flow for Cooking Control Function



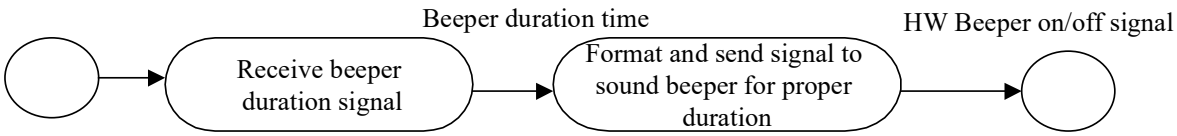
Data Flow for Oven Control Function



Data Flow for Display Time Function

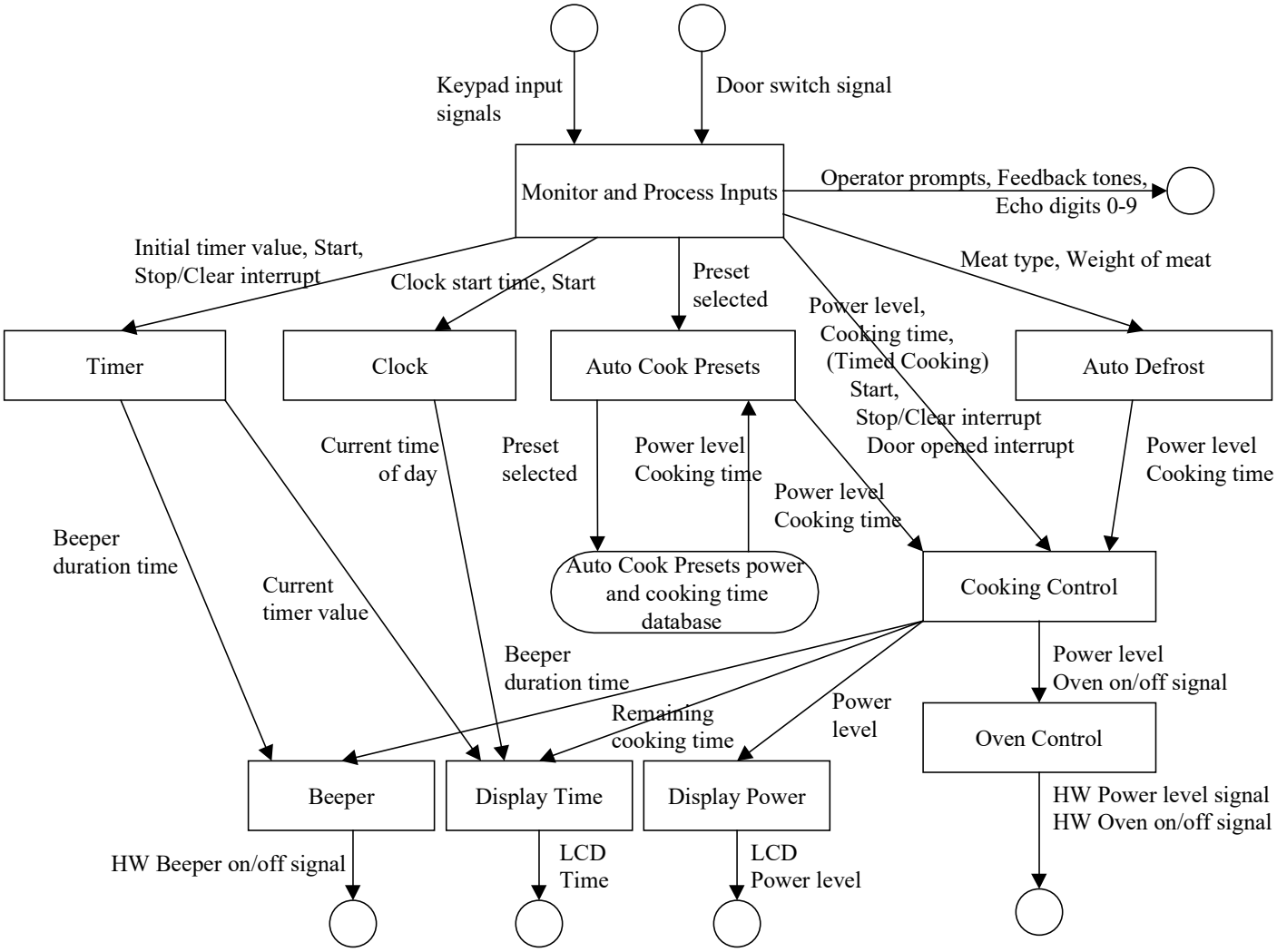


Data Flow for Display Power Level Function



Data Flow for Beeper Function

8.2.2 Final Structure Chart for All Major Functions



Final Structure Chart for Microwave Oven

8.2.3 Data Dictionary

Entity Name	Type	Description
Auto Cook Presets	FUNCTION	Looks up proper power level and cooking time settings for a given auto cook preset food item
Auto Defrost	FUNCTION	Determines correct power level and cooking time values based on type of meat and weight of meat being defrosted
Beeper	FUNCTION	Sounds beeper tone for specified length of time
Beeper duration time	INTEGER	The number of seconds that the beeper is to sound a tone.
Clock	FUNCTION	Begins measuring time from an initial time of day input
Clock start time	TIME_H_M	Operator entered time value used to initialize the clock
Cooking Control	FUNCTION	Sends signals to set oven control function to set proper power level then turn oven on for the length of time specified in the input to this function
Current time of day	TIME_H_M	Current time of day to be displayed
Current timer value	TIME_M_S	Current timer value to be displayed
Display Power	FUNCTION	Sends power level setting to LCD screen to be displayed
Display Time	FUNCTION	Sends time setting to LCD screen to be displayed
Door opened interrupt	ENUMERATION	Door closed = 0, Door open = 1
Door switch signal	DOOR_SWITCH	Signal from door sensor to indicate whether the microwave door is opened or closed
Echo digits 0-9	ECHO_DIGITS	Echo digits 0-9 to LCD screen as operator presses these keys
Feedback tones	FEEDBACK_TONES	Sound tones after each key is entered
HW Beeper on off signal	HW_BEEPER_CNTL	Discrete signal to beeper hardware to turn beeper on and off
HW Oven on off signal	HW_OVEN_CNTL	Discrete signal to microwave generator hardware to turn oven on and off
HW Power level	HW_PWR_LEVEL	Signal to set microwave generator hardware to correct value

Initial timer value	TIME_M_S	The operator entered time value used to initialize the timer
Keypad input signals	KEYPAD_INPUTS	Input signals from keypad hardware
LCD Power level	LCD_PWR_LEVEL	Power level formatted for display on LCD screen.
LCD Time	LCD_TIME	Time value (XX:XX) sent to LCD screen (can be either in hours, minutes or in minutes, seconds depending on which microwave function is active)
Meat type	ENUMERATION	Type of meat (Beef = 1, Poultry = 2, Fish = 3)
Monitor and Process Inputs	FUNCTION	Receives and decodes inputs from keypad and door sensor then calls appropriate sub-functions
Operator prompts	OPER_PROMPT	Operator prompts sent to the LCD screen to request operator inputs
Oven on/off signal	ENUMERATION	Off = 0, On = 1
Oven Control	FUNCTION	Sends power level settings to microwave generator hardware then turns microwave generator hardware on and off as inputs indicate
Power level	INTEGER	Power level selected by operator (1 – 10)
Preset selected	ENUMERATION	Preset key selected by operator (Popcorn = 0, Potato = 1, Beverage = 2, Pizza = 3, Frozen Entrée = 4, Fresh Veg. = 5)
Remaining cooking time	TIME_M_S	Length of time before cooking completes
Start	ENUMERATION	Indicates that the Start button has been pushed (Start active = 1, Start inactive = 0)
Stop/Clear interrupt	ENUMERATION	Stop/Clear button indication signal (Stop/Clear active = 1, Stop/Clear inactive = 0)
Timer	FUNCTION	Receives an initial timer value then counts down the time to 00:00 and sends a signal to the beeper function to sound a tone.
Weight of meat	FLOAT	Weight of meat for defrosting in ounces

8.2.4 Interface Definition

Note: Parameters are listed by Name, Type, and In/Out/In-Out to indicate if they are read-only (In), write-to (Out), or read-and-write (In-Out). Types that are listed in all caps are variable types that have been defined for this specific application. See Data Dictionary for a further breakdown of ENUMERATION types.

8.2.4.1 Monitor and Process Inputs Function

Function Name:

Monitor and Process Inputs (see note in section 8.2)

Parameters:

Door_switch_signal, DOOR_SWITCH, In
Keypad_input_signals, KEYPAD_INPUTS, In
Clock_start_time, TIME_H_M, Out
Cooking_time, TIME_M_S, Out
Door_opened_interrupt, ENUMERATION, Out
Echo_digits_0_9, ECHO_DIGITS, Out
Feedback_tones, FEEDBACK_TONES, Out
Initial_timer_value, TIME_M_S, Out
Meat_type, MEAT_TYPE, Out
Operator_prompts, OPER_PROMPT, Out
Power_level, Integer, Out
Preset_selected, ENUMERATION, Out
Start, ENUMERATION, Out
Stop_Clear_Interrupt, ENUMERATION, Out
Weight_of_meat, Float, Out

Purpose:

This function receives inputs from the microwave keypad and from the microwave door sensor. Beeper tones are sounded after each keypad button is pressed and digits 0 through 9 are echoed to the LCD screen. Keypad inputs are decoded and the proper sub-functions are called depending on what operator inputs are received. If the Auto Defrost button input is received this function sends a prompt to the LCD display to request that the operator input the type and weight of meat to be defrosted. If Power Level button input is received this function sends a prompt to the LCD display requesting that the operator input a power level from 1 to 10. If a door open signal is received, and the Cooking Function is active, this function sends a Stop Interrupt signal to the Cooking Function. If the Start button input is then received, and the door signal indicates that the microwave door has been closed, this function will send a Start signal to

the Cooking Function to restart the oven. Likewise, if the Stop/Clear button is pressed while the Timer Function is active this function will send a Stop Interrupt signal to the Timer Function to pause the timer countdown. If the Start button input is then received this function will send a Start signal to the Timer function to restart the timer countdown.

Traceability:

<See Requirement 5.3.1.1>
<See Requirement 5.3.1.2>
<See Requirement 5.3.1.3>
<See Requirement 5.3.1.4>
<See Requirement 5.3.1.5>
<See Requirement 5.3.2.1>
<See Requirement 5.3.2.2>
<See Requirement 5.3.2.3>
<See Requirement 5.3.2.5>
<See Requirement 5.3.2.6>
<See Requirement 5.3.3.1>
<See Requirement 5.3.3.2>
<See Requirement 5.3.3.3>
<See Requirement 5.3.3.7>
<See Requirement 5.3.3.8>
<See Requirement 5.3.3.9>
<See Requirement 5.3.3.10>
<See Requirement 5.3.4.1>
<See Requirement 5.3.4.3>
<See Requirement 5.3.5.1>
<See Requirement 5.3.5.2>
<See Requirement 5.3.5.3>
<See Requirement 5.3.5.4>
<See Requirement 5.3.5.5>
<See Requirement 5.3.5.6>
<See Requirement 5.3.6.1>
<See Requirement 5.3.6.2>
<See Requirement 5.3.6.3>
<See Requirement 5.3.6.4>
<See Requirement 5.3.6.8.a>
<See Requirement 5.3.6.12>

8.2.4.2 Clock Function

Function Name:

Clock

Parameters:

Clock_start_time, TIME_H_M, In
Start, ENUMERATION, In
Current_time_of_day, TIME_H_M, Out

Purpose:

The clock function receives an initial time of day setting from the operator then increments the current time value once per minute. Once a minute the current time value is output to the Display Time function for display.

Traceability:

<See Requirement 5.3.2.4>
<See Requirement 5.3.2.6>
<See Requirement 5.3.2.7>
<See Requirement 5.3.2.8>
<See Requirement 5.3.2.9>

8.2.4.3 Timer Function

Function Name:

Timer

Parameters:

Initial_timer_value, TIME_M_S, In
Start, ENUMERATION, In
Stop_Clear_Interrupt, ENUMERATION, In
Beeper_duration_time, Integer, Out
Current_timer_value, TIME_M_S, Out

Purpose:

This function receives an initial timer value that was input from the operator then counts down the time until it reaches zero. As the timer counts down the current time is sent once per second to the Display Time function. Once the timer reaches zero a value is sent to the Beeper function indicating the length of time that the beeper should sound. If a Stop interrupt signal is received during the count down, the time is halted until either a Stop/Clear signal is received (to cancel the timer countdown) or a Start signal is received (to continue the timer countdown).

Traceability:

<See Requirement 5.3.3.4>
<See Requirement 5.3.3.5>
<See Requirement 5.3.3.6>
<See Requirement 5.3.3.7>
<See Requirement 5.3.3.8>

8.2.4.4 Auto Cook Presets Function

Function Name:
Auto Cook Presets

Parameters:
Preset_selected, ENUMERATION, In
Cooking_time, TIME_M_S, Out
Power_level, Integer, Out

Purpose:
This function receives an input indicating which operator preset button was pressed. The predefined power level and cooking time associated with that preset button are looked up in the Auto Cook Preset power and cooking time database. The power and cooking time values are output to the Cooking Control Function.

Traceability:
<See Requirement 5.3.4.2>
<See Requirement 5.3.4.3.a>

8.2.4.5 Auto Defrost Function

Function Name:
Auto Defrost

Parameters:
Meat_type, MEAT_TYPE, In
Weight_of_meat, Float, In
Cooking_time, TIME_M_S, Out
Power_level, Integer, Out

Purpose:
This function receives a type of meat and the weight of the meat from the Monitor and Process Inputs function. Based on the type of meat and the weight of meat the function determines the correct values for power level and cooking time. The power level and cooking time values are then output to the Cooking Control Function.

Traceability:
<See Requirement 5.3.5.6>

8.2.4.6 Cooking Control Function

Function Name:
Cooking Control

Parameters:
Cooking_time, TIME_M_S, In
Door_opened_interrupt, ENUMERATION, In
Start, ENUMERATION, In
Stop_Clear_Interrupt, ENUMERATION, In
Power_level, Integer, In-Out
Beeper_duration_time, Integer, Out
Oven_on_off_signal, ENUMERATION, Out
Remaining_cooking_time, TIME_M_S, Out

Purpose:
The Cooking Control Function power level and cooking time inputs then passes on the correct power level to the Oven Control Function and turns the oven on for the correct length of time. If a Stop interrupt signal is received during cooking this function must send an Oven off signal to the Oven Control Function. If a Start signal is then received the function sends an Oven on signal to the Oven Control Function for the remaining amount of cooking time.

Traceability:
<See Requirement 5.3.4.3.b>
<See Requirement 5.3.6.5>
<See Requirement 5.3.6.6>
<See Requirement 5.3.6.7>
<See Requirement 5.3.6.8>
<See Requirement 5.3.6.9>
<See Requirement 5.3.6.10>
<See Requirement 5.3.6.11>

8.2.4.7 Oven Control Function

Function Name:
Oven Control

Parameters:
Power_level, Integer, In
Oven_on_off_signal, ENUMERATION, In-Out
HW_Oven_on_off_signal, HW_OVEN_CNTL, Out
HW_Power_level, HW_PWR_LEVEL, Out

Purpose:

The Oven Control function receives power level and oven on/off inputs from the cooking function then sends signals to the microwave generator hardware to set the correct oven power setting then turn on and off the oven as commanded.

Traceability:

<See Requirement 5.3.6.6>
<See Requirement 5.3.6.7.a>
<See Requirement 5.3.6.11.a>

8.2.4.8 Display Time Function

Function Name:

Display Time

Parameters:

Current_time_of_day, TIME_H_M, In
Current_timer_value, TIME_M_S, In
Remaining_cooking_time, TIME_M_S, In
LCD_Time, LCD_TIME, Out

Purpose:

This function receives a time input that can come from the clock function, the timer function, or the cooking control function. This time value is formatted correctly and then output to the LCD screen hardware for display.

Traceability:

<See Requirement 5.3.2.4>
<See Requirement 5.3.2.7>
<See Requirement 5.3.3.6>
<See Requirement 5.3.4.3.b>
<See Requirement 5.3.6.5>
<See Requirement 5.3.6.7.c>
<See Requirement 5.3.6.8.d>

8.2.4.9 Display Power Function

Function Name:

Display Power

Parameters:

Power_level, integer, In

LCD_Power_level, LCD_PWR_LEVEL, Out

Purpose:

This function receives an integer indicating the current microwave power setting. This power setting value is formatted correctly and then output to the LCD screen hardware for display.

Traceability:

<See Requirement 5.3.6.1>

8.2.4.10 Beeper Function

Function Name:

Beeper

Parameters:

Beeper_duration_time, Integer, In
HW_Beeper_on_off_signal, HW_BEEPER_CNTL, Out

Purpose:

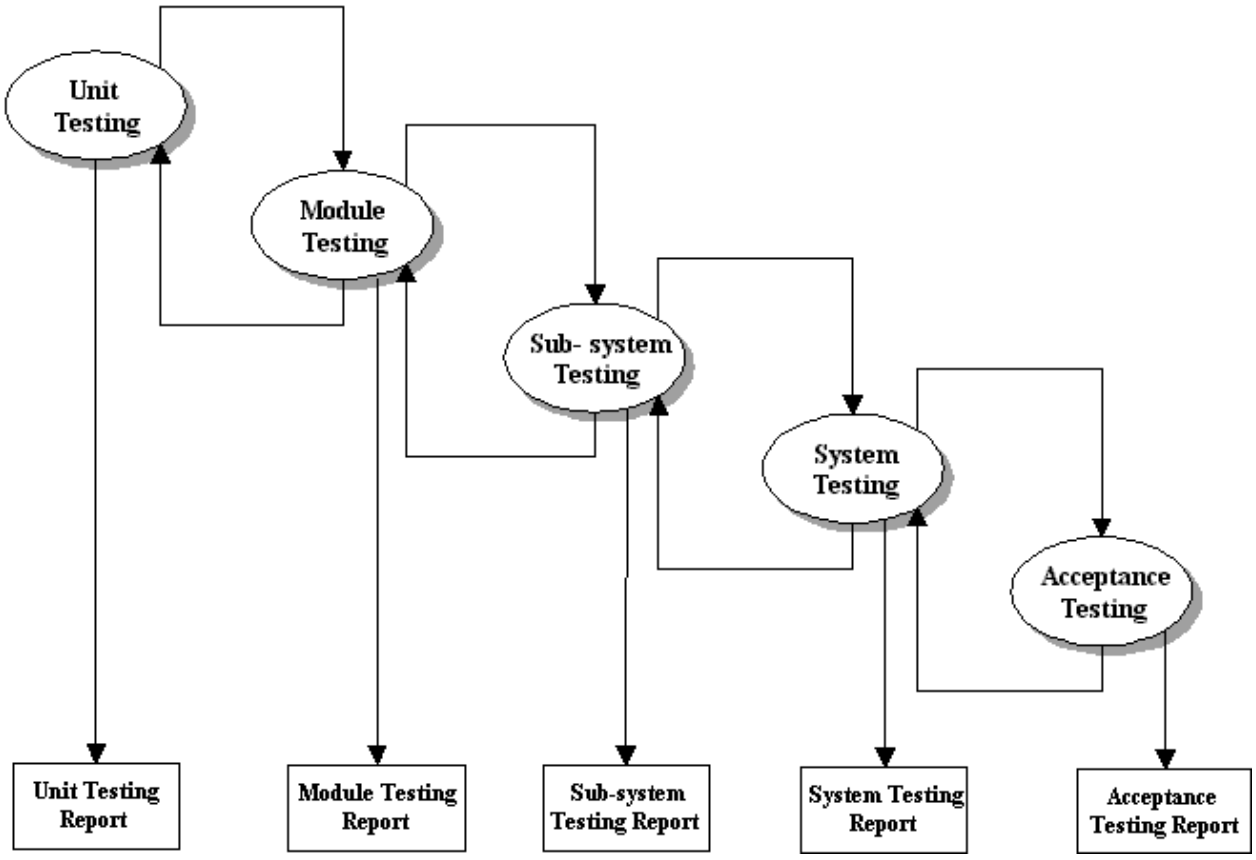
This function receives an integer that specifies the length of time in seconds that the microwave's beeper should sound a tone. The function sends a discrete signal to the beeper hardware to turn tone on then turn tone off after the specified length of time has passed.

Traceability:

<See Requirement 5.3.1.1>
<See Requirement 5.3.1.2>
<See Requirement 5.3.3.5>
<See Requirement 5.3.3.9>
<See Requirement 5.3.6.6>
<See Requirement 5.3.6.8.b>
<See Requirement 5.3.6.10.b>
<See Requirement 5.3.6.11.b>

9 TESTING

9.1 TESTING PROCESS



Component Testing – Static Integration Testing – Dynamic Environment Testing – Dynamic

Each test phases shown above is described in more detail in the following sections.

9.1.1 Unit Testing (Static)

- Involves testing the components of each module to insure that they were correctly implemented. Program inspections will be used to detect software defects. A checklist of likely errors will be developed to drive the inspection process. Teams of four to six software developers will be responsible for performing the inspections.

9.1.2 Module Testing (Dynamic)

- Input Monitoring and Processing Testing: Tests for accuracy in decoding input from keypad and door sensor. Also tests proper calling of subfunctions based on input from keypad and door sensor.

- Clock Testing: Tests for errors in the display of the time on the LCD. Tests for proper numeric incrementation.
- Timer Testing: Tests for accurate timer operation based on inputs received from the keypad. Checks for errors in decrementing the time displayed on the LCD. Also tests whether timer is stopped when stop/clear key is pressed.
- Auto Cook Presets Testing: Tests for accuracy in determining the proper power level and cooking time to be set depending on which preset key is pressed.
- Auto Defrost Testing: Tests for errors in the setting of power levels and defrosting times depending on the food product that is being defrosted.
- Cooking Control Testing: Tests for errors in reception of time and power level inputs and checks the accuracy of signals to the oven indicating power levels and cooking duration. Also tests whether a cooking process stops if a signal interrupt originating from the stop/clear key is received.
- Oven Control Testing: Tests for errors in the control of the microwave generator. Assures proper power settings are used and the generator turns off and on as directed by the cooking control function.
- Time Display Testing: Checks for errors in time formatting and output on the LCD.
- Power Display Testing: Checks for errors in the display and formatting of power level settings on the LCD.
- Beeper Testing: Checks for errors in sound tone and duration given signals to the beeper from control software.

Note: See Interface Definition (section 8.2.4) above for a detailed listing of the specific requirements that will be tested for each of the modules listed above.

9.1.3 Sub-system Integration Testing (Dynamic)

- Command Processing System Testing: Tests for errors in integration and functioning of system sub-components. Assures proper operation of Input Monitoring and Processing, Clock, Timer, Auto Cook, and Auto Defrost functions.
- Cooking Control System Testing: Tests for errors in integration and functioning of cooking control function and oven control function.
- Output Processing System: Tests for errors in integration of timer display, clock display, and beeper signaling.

9.1.4 System Testing (Dynamic)

- The system testing procedure tests for errors resulting from unexpected interactions among sub-systems and system components. The procedure tests for the accurate processing of input by the Command Processing System. The Command Processing System must also send correct signals and information

to the other sub-systems based on this input. The Oven Control System must correctly receive signals and information from the Command Processing System and send the appropriate signals to the microwave generator. Likewise, the Output Processing System must also correctly accept signals and information from the Command Processing System and send appropriate signals to the LCD and beeper mechanism. The hardware components must all correctly accept signals from their assigned sub-system.

9.1.5 Acceptance Testing (Dynamic)

- Acceptance testing will be performed based on inputs and conditions that would normally occur in an environment in which the microwave oven would typically be used. These tests will be performed for final approval before the manufacturing of the production units.

9.2 TESTING SCHEDULE

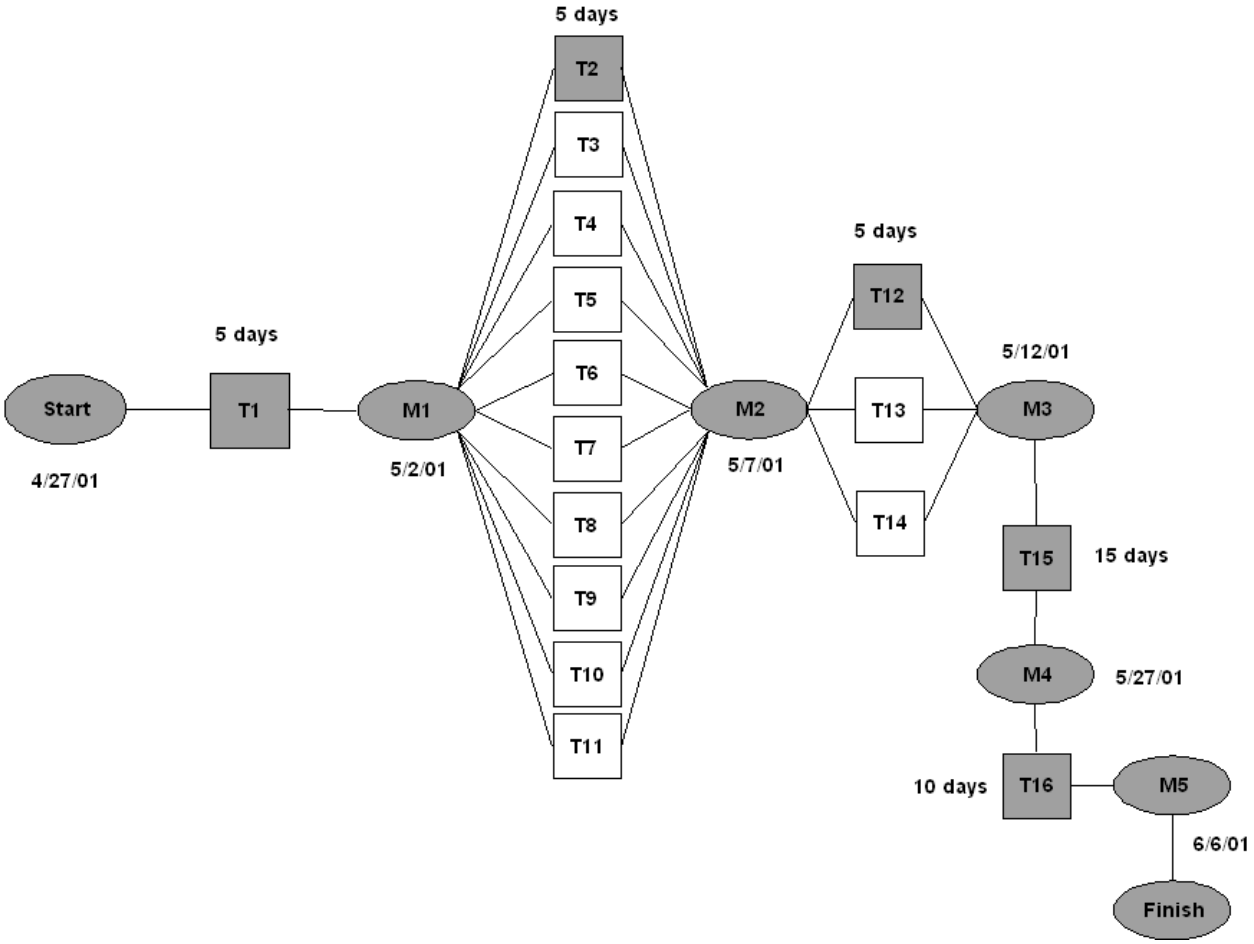
9.2.1 Task Duration and Dependency Table

Task	Description	Duration in Work Days	Dependent Upon
T1	Individual Component Testing	5 days	None
T2	Input Processing and Monitoring Testing	5 days	T1
T3	Clock Module Testing	1 day	T1
T4	Timer Module Testing	1 day	T1
T5	Auto Cook Preset Module Testing	1 day	T1
T6	Auto Defrost Module Testing	1 day	T1
T7	Cooking Control Module Testing	2 days	T1
T8	Oven Control Module Testing	1 day	T1
T9	Timer Display Module Testing	1 day	T1
T10	Power Display Module Testing	1 day	T1
T11	Beeper Module Testing	1 day	T1
T12	Command Processing Sub-system Testing	5 days	T2 – T11
T13	Oven Control Sub-system Testing	4 days	T2 – T11
T14	Output Processing Sub-system Testing	2 days	T2 – T11
T15	Microwave Oven System Testing	15 days	T12 – T14
T16	Acceptance Testing	10 days	T15

9.2.2 Personnel Allocation Table

Task	Description	Personnel
Task 1	Individual Component Testing	D. David, J. Green, A. Largent
Task 2	Input Processing and Monitoring Testing	D. David, A. Largent
Task 3	Clock Module Testing	J. Green
Task 4	Timer Module Testing	A. Largent
Task 5	Auto Cook Preset Module Testing	D. David
Task 6	Auto Defrost Module Testing	J. Green
Task 7	Cooking Control Module Testing	A. Largent
Task 8	Oven Control Module Testing	D. David
Task 9	Timer Display Module Testing	J. Green
Task 10	Power Display Module Testing	A. Largent
Task 11	Beeper Module Testing	J. Green
Task 12	Command Processing Sub-system Testing	D. David, J. Green
Task 13	Oven Control Sub-system Testing	J. Green, A. Largent
Task 14	Output Processing Sub-system Testing	D. David, J. Green
Task 15	Microwave Oven System Testing	D. David, J. Green, A. Largent
Task 16	Acceptance Testing	D. David, J. Green, A. Largent

9.2.3 Activity Network



Shaded areas indicate Critical Path
Length of Critical Path: 40 Days

Note: See Task Duration and Dependency Table in section 9.2.1 for the duration of tasks that are not on the critical path.

Milestone M1: A report will be generated following the Unit Testing Process. Completion of Unit Test Report indicates that the milestone has been reached.

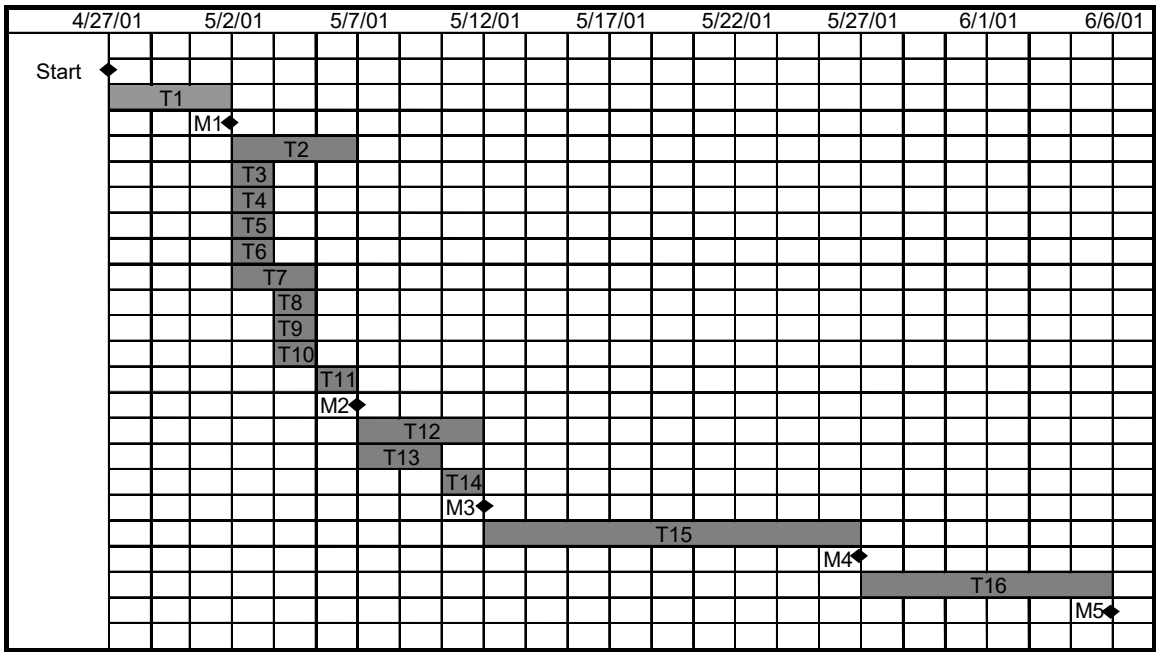
Milestone M2: A report will be generated at the conclusion of the Module Testing Process. Completion of the Module Test Report indicates the milestone has been reached.

Milestone M3: A report will be generated at the conclusion of Sub-system testing. Completion of the Sub-system Test Report indicates the milestone has been reached.

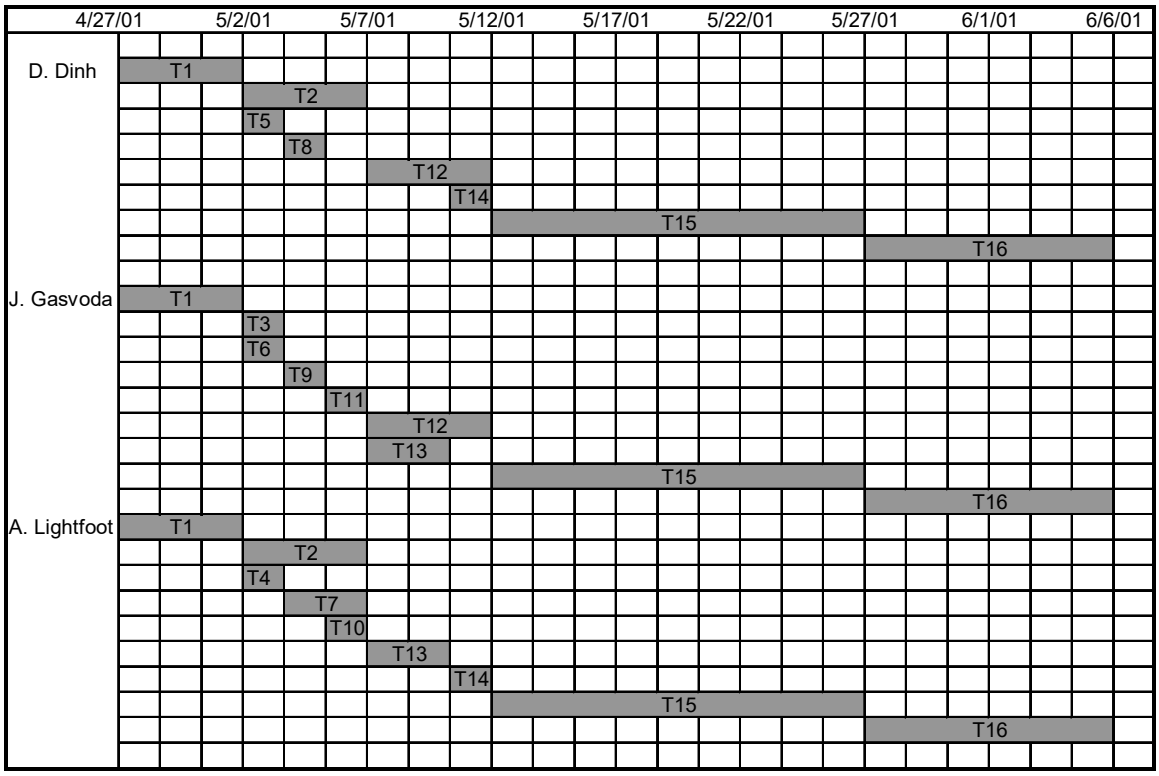
Milestone M4: A report will be generated at the conclusion of System testing. Completion of the System Test Report indicates the milestone has been reached.

Milestone M5: A report will be generated at the conclusion of the Acceptance Testing process. Completion of the Acceptance Test Report indicates that the final milestone has been completed.

9.2.4 Activity Schedule



9.2.5 Personnel Allocation versus Time



9.3 TESTING RECORDING PROCEDURE

A Test Results Table will be filled in as each test is executed (see example table below). The actual results will be compared with the expected results for each test step. If the actual and expected results match, a "P" will be placed in the Pass/Fail column. If the actual and expected results do not match, an "F" will be placed in the Pass/Fail column and a Problem Report Number will be assigned to track the problem. See the Problem Report Tracking Table below in section 9.3.2.

9.3.1 Test Results Table

Test	Step	Action	Expected Result	Actual Result	Pass/Fail	Problem Report Number
Test 1	Step 1	Set cooking time to 00:30 and push Start button.	Oven comes on and the displayed time counts down for 30 seconds to 00:00 at which time the oven shuts off and the beeper sounds	Beeper did not sound when time reached 00:00.	F	5
Test 2	Step 1	Set cooking time to 00:30 and push Start button.	Oven turns on and displayed time begins decrementing from 00:30.	Same as expected result.	P	N/A
	Step 2	When time equals 00:15 push Stop/Clear button.	Oven shuts off with 00:15 displayed on LCD screen.	Same as expected result.	P	N/A
	Step 3	Push Start button.	Oven turns back on and time counts down to 00:00 at which time the oven shuts off and the beeper sounds	Oven failed to turn on when Start button was pressed.	F	6

9.3.2 Problem Report Tracking Table

All problem reports that are written during the various test phases will be tracked in the table below. This table records the PR number, who originated the report, the date that it was originated, a description of the problem, the test phase where the problem was found, the type of problem that was found, the person who is assigned to work the problem, and the current status of each problem report.

Problem Number	Origination Date	Originator	Problem Description	Where Found (1)	Problem Type (2)	Assignee	Status (3)
1	4/30/01	D. David	Oven remains on when oven door is opened.	U	C	D. David	Open
2	5/2/01	J. Green	Auto defrost power level incorrect for popcorn off	M	C	D. David	Open
3	5/5/01	A. Largent	Invalid power level of 0 was not handled properly.	M	C	A. Largent	Open
4	5/16/01	D. David	Beeper failed to sound when Auto cook button was pressed	S	HW	D. David	Open
5	5/24/01	J. Green	Beeper did not sound when time reached 00:00	S	HW	J. Green	Open
6	5/27/01	A. Largent	Oven failed to turn on when Start button was pressed	S	C	J. Green	Open

(1) Where Found:
- U: Unit Test
- M: Module Test
- SS: Sub-system Test
- S: System Test
- A: Acceptance Test
- P: Post-acceptance Test
- O: Other

(2) Problem Type:

RD: Requirements Definition
RS: Requirements Specification
C: Code
UM: User's Manual
HW: Hardware
O: Other

(3) Status:

O: Open
C: Closed

9.4 HARDWARE AND SOFTWARE REQUIREMENTS

Each person involved in testing software for this project will require at least a personal computer that meets the minimum hardware and software requirements listed below:

9.4.1 Hardware Requirements for Testing

Hardware	Status
Pentium 3, 550 MHz processor	On-hand
Windows 98 Operating System	On-hand
At least 64 MB RAM	On-hand
Monitor	On-hand
Mouse	On-hand
Keyboard	On-hand
In Circuit Emulator	To be purchased
HP 1631A Logic Analyzer	To be purchased
Digital Multimeter	To be purchased
Network server for storing files under configuration control	On-hand

9.4.2 Software Requirements for Testing

Software	Status
Microsoft Visual Test (for automation test)	On-hand
Microsoft Office 2000 - Professional	On-hand
Netscape Communicator (Version 4.61 or later) or Microsoft Internet Explorer (Version 5.0 or later)	On-hand
ZoneAlarm firewall software	To be purchased

9.5 TRACEABILITY TO REQUIREMENTS

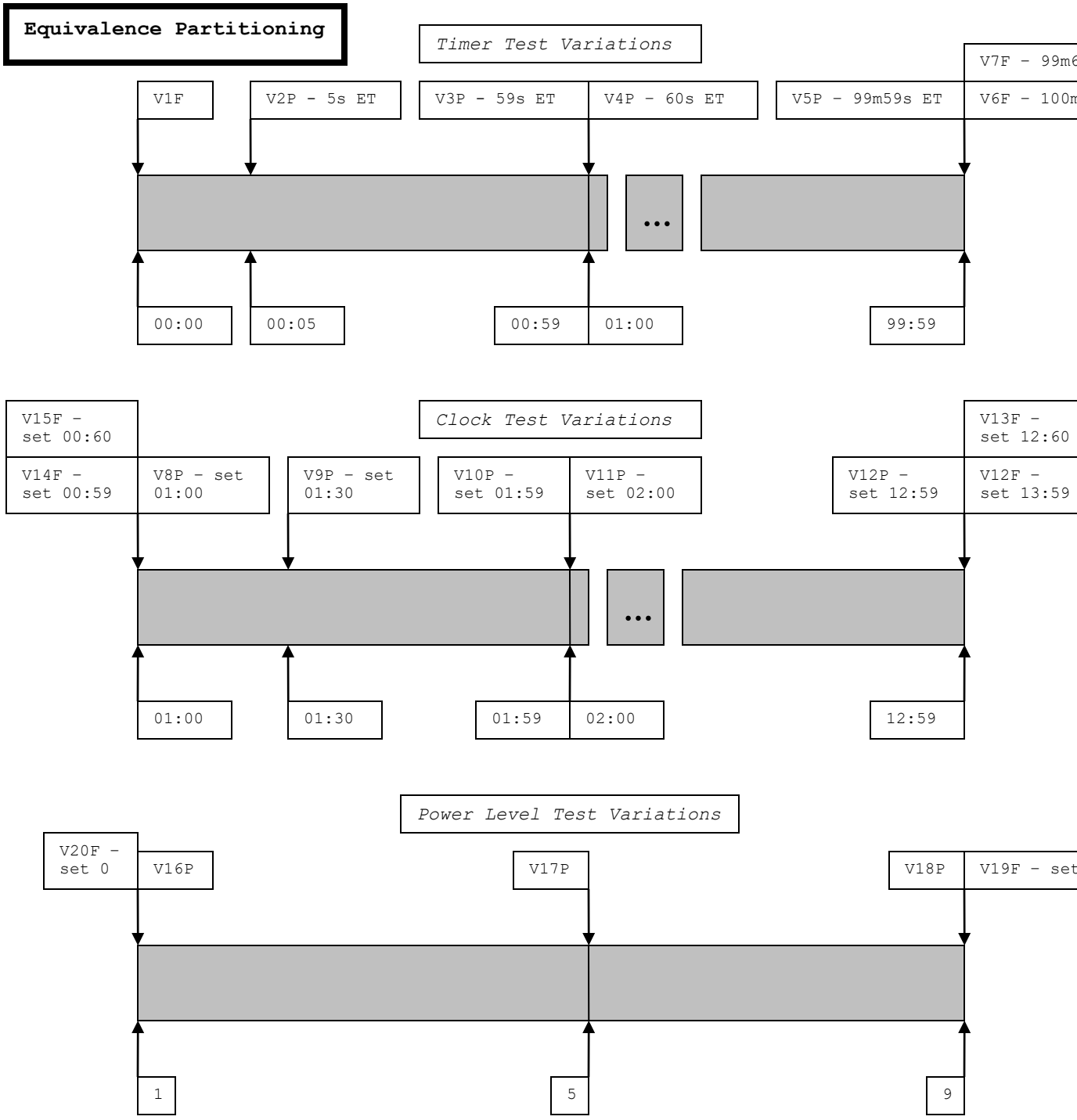
Task	Description	Requirements Tested
Task 1	Unit testing – Test individual components of each module	All requirements statically tested.
Task 2	Monitor and Process Inputs Module	5.3.1.1–5.3.1.5, 5.3.2.1–5.3.2.6, 5.3.3.1–5.3.3.3, 5.3.3.7–5.3.3.10, 5.3.4.1, 5.3.4.3, 5.3.5.1–5.3.5.6, 5.3.6.1–5.3.6.4, 5.3.6.8.a, and 5.3.6.12.
Task 3	Clock Module	5.3.2.4, and 5.3.2.6–5.3.2.9
Task 4	Timer Module	5.3.3.4–5.3.3.8
Task 5	Auto Cook Presets Module	5.3.4.2, and 5.3.4.3.a
Task 6	Auto Defrost Module	5.3.5.6
Task 7	Cooking Control Module	5.3.4.3.b, and 5.3.6.5–5.3.6.11
Test 8	Oven Control Module	5.3.6.6, 5.3.6.7.a, and 5.3.6.11.a
Task 9	Display Time Module	5.3.2.4, 5.3.2.7, 5.3.3.6, 5.3.4.3.b, 5.3.6.5, 5.3.6.7.c, and 5.3.6.8.d
Task 10	Display Power Module	5.3.6.1
Task 11	Beeper Module	5.3.1.1, 5.3.1.2, 5.3.3.5, 5.3.3.9, 5.3.6.6, 5.3.6.8.b, 5.3.6.10.b, and 5.3.6.11.b
Task 12	Command Processing Sub-system	5.3.1.1–5.3.1.5, 5.3.2.1–5.3.2.6, 5.3.3.1–5.3.3.10, 5.3.4.1, 5.3.4.3, 5.3.5.1–5.3.5.6, 5.3.6.1–5.3.6.4, 5.3.6.8.a, and 5.3.6.12.
Task 13	Oven Control Sub-system	5.3.4.3.b, and 5.3.6.5–5.3.6.11
Task 14	Output Processing Sub-system	5.3.1.1, 5.3.1.2, 5.3.2.4, 5.3.2.7, 5.3.3.5, 5.3.3.6, 5.3.3.9, 5.3.4.3.b, 5.3.6.1, 5.3.6.5, 5.3.6.6, 5.3.6.7.c, 5.3.6.8.b, 5.3.6.8.d, 5.3.6.10.b, and 5.3.6.11.b
Task 15	Microwave Oven System	All requirements dynamically tested.
Task 16	Acceptance Testing	To be filled in when acceptance test procedures are developed.

The table above traces requirements to the various tasks that are being performed during the different stages of testing. Further traceability will be provided in the headers of each test procedure. Each test case header is required to contain a section titled “Requirements tested:”. In this section all software requirements being tested

will be listed and enclosed in greater-than, less-than brackets (e.g. <5.3.5.6>). A tool will be written to extract the requirements traceability information from all test cases and sort this information by requirement number. The tool will capture the name of the procedure from which the traceability information was extracted. The sorted output of the tool will clearly show which test or tests were used to verify which requirements. The sorted output will also show whether or not there were any requirements that were "missed" (not tested by any of the test procedures). This process for requirements traceability will be added to our Software Test Plan.

9.6 BLACK BOX TESTING

9.6.1 Equivalence Partitions



9.6.2 Black Box Test Case Table

Test	Action	Expected Result	Actual Result	Pass/ Fail	Problem Report Number
Test 1 Cooking Time					
Test 1.1 Boundary Value	Set cooking time to 00:00 and push Start button.	Oven remains off, displayed time is 00:00, and the beeper sounds.			
Test 1.2 Typical Value	Set cooking time to 00:30 and push Start button.	Oven turns on and displayed time begins decrementing from 00:30. Time decrements once per second until displayed time is 00:00 at which time the oven shuts off and the beeper sounds.			
Test 1.3 Boundary Value	Set cooking time to 00:59 and push Start button.	Oven turns on and displayed time begins decrementing from 00:59. Time decrements once per second until displayed time is 00:00 at which time the oven shuts off and the beeper sounds.			
Test 1.4 Boundary Value	Set cooking time to 01:00 and push Start button.	Oven turns on and displayed time begins decrementing from 01:00. Time decrements once per second until displayed time is 00:00 at which time the oven shuts off and the beeper sounds.			
Test 1.5 Typical Value	Set cooking time to 03:25 and push Start button.	Oven turns on and displayed time begins decrementing from 03:25. Time decrements once per second until displayed time is 00:00 at which time the oven			

		shuts off and the beeper sounds.			
Test 1.6 Boundary Value	Set cooking time to 99:59 and push Start button.	Oven turns on and displayed time begins decrementing from 99:59. Time decrements once per second until displayed time is 00:00 at which time the oven shuts off and the beeper sounds.			
Test 2 Timer					
Test 2.1 Boundary Value	Set timer to 00:00 and push Start button.	Oven remains off, Displays shows 00:00 and beeper immediately sounds.			
Test 2.2 Typical Value	Set timer to 00:30 and push Start button.	Oven remains off and displayed time begins decrementing from 00:30. Time decrements once per second until displayed time is 00:00 at which time the beeper sounds.			
Test 2.3 Boundary Value	Set timer to 00:59 and push Start button.	Oven remains off and displayed time begins decrementing from 00:59. Time decrements once per second until displayed time is 00:00 at which time the beeper sounds.			
Test 2.4 Boundary Value	Set timer to 01:00 and push Start button.	Oven remains off and displayed time begins decrementing from 01:00. Time decrements once per second until displayed time is 00:00 at which time the beeper sounds.			
Test 2.5 Typical Value	Set timer to 49:50 and push Start button.	Oven remains off and displayed time begins decrementing from 49:50. Time decrements once per second until			

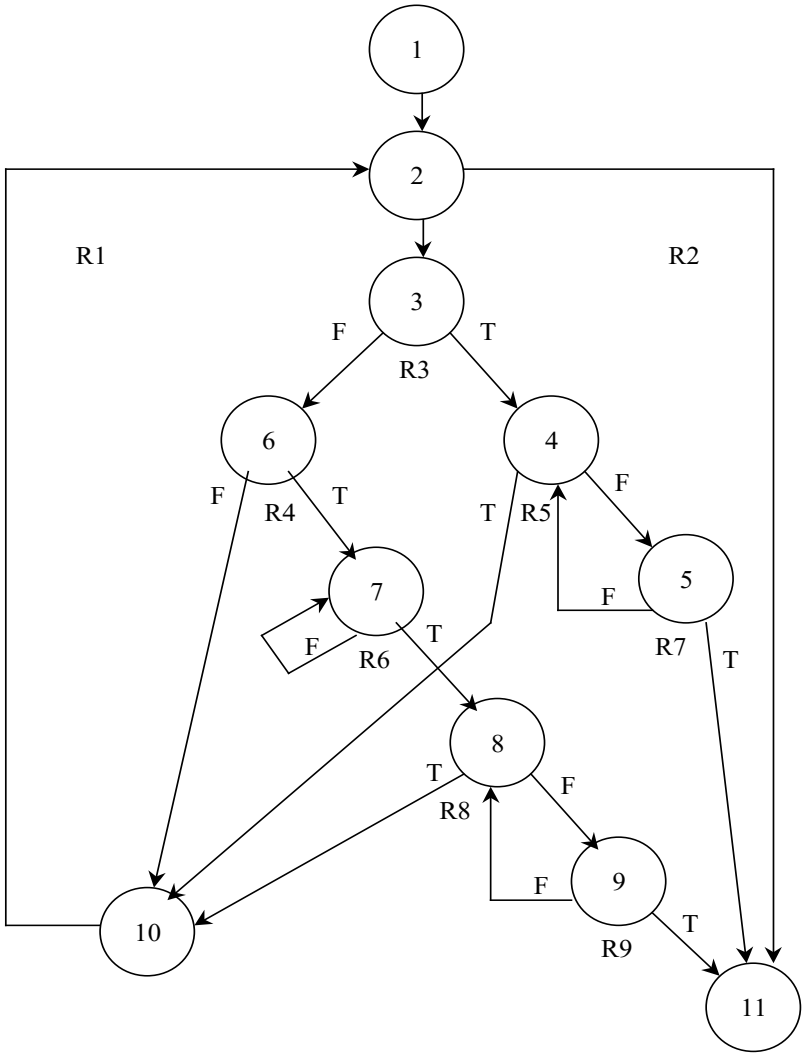
		displayed time is 00:00 at which time the beeper sounds.			
Test 2.6 Boundary Value	Set timer to 99:59 and push Start button.	Oven remains off and displayed time begins decrementing from 99:59. Time decrements once per second until displayed time is 00:00 at which time the beeper sounds.			
Test 3 Clock					
Test 3.1 Invalid Value	Set clock to 00:00 and push Start button.	Beeper sounds and display indicates that an invalid time value was entered.			
Test 3.2 Boundary Value	Set clock to 01:00 and push Start button.	Time of day displayed begins incrementing once per second from 01:00.			
Test 3.3 Typical Value	Set clock to 01:30 and push Start button.	Time of day displayed begins incrementing once per second from 01:30.			
Test 3.4 Boundary Value	Set clock to 01:59 and push Start button.	Time of day displayed increments from 01:59 to 02:00 then continues incrementing once per second.			
Test 3.5 Boundary Value	Set clock to 02:00 and push Start button.	Time of day displayed begins incrementing once per second from 02:00..			
Test 3.6 Typical Value	Set clock to 05:30 and push Start button.	Time of day displayed begins incrementing once per second from 05:30.			
Test 3.7 Boundary Value	Set clock to 12:59 and push Start button.	Time of day displayed increments from 12:59 to 01:00 then continues incrementing once per second			
Test 3.8 Invalid Value	Set clock to 13:00 and push Start button.	Beeper sounds and display indicates that an invalid time value was			

		entered.			
Test 4 Power Level					
Test 4.1 Invalid Value	Set Power Level to 0, set time to 00:30, and push Start button.	Beeper sounds and display indicates that an invalid power level value was entered.			
Test 4.2 Boundary Value	Set Power Level to 1, set time to 00:30, and push Start button.	Oven turns on, time begins decrementing from 00:30, and the display shows that power level 1 has been selected.			
Test 4.3 Typical Value	Set Power Level to 5, set time to 00:30, and push Start button.	Oven turns on, time begins decrementing from 00:30, and the display shows that power level 5 has been selected.			
Test 4.4 Boundary Value	Set Power Level to 10, set time to 00:30, and push Start button.	Oven turns on, time begins decrementing from 00:30, and the display shows that power level 10 has been selected.			
Test 4.5 Invalid Value	Set Power Level to 11, set time to 00:30, and push Start button.	Beeper sounds and display indicates that an invalid power level value was entered.			

9.7 STRUCTURAL (WHITE BOX TESTING) METHOD

9.7.1 Flow Graph for Cooking Control Function

```
Begin Cooking Control Function
1.   initialize time and power values
2.   while time is not equal to 00:00
3.      if Stop/Clear button is pressed
           call Oven Control Function
              to shut off oven
4.        while Start button is not pressed
5.           if Stop/Clear is pressed again
                end cooking
             endif
           endwhile
6.      elseif microwave door is opened
           call Oven Control Function
              to shut off oven
7.        while door is open
             wait
           endwhile
8.        while Start button is not pressed
9.           if Stop/Clear is pressed
                end cooking
             endif
           endwhile
        endif
10.   call Display Time Function
        to output current time value
      call Display Power Function to
        display current power level
      call Oven Control Function to turn
        on oven at the proper power level
      decrement current time value
    endwhile
11. call Oven Control Function to
      shut off oven
    set time to 00:00
    call Display Time Function
    call Beeper Function to sound beeper
End Cooking Control Function
```



9.7.2 Regions in Flow Graph

Nine regions were identified in the flow graph for the Cooking Control Function. These regions are labeled R1 through R9 on the flow graph in section 9.7.1.

9.7.3 Cyclomatic Complexity

The value computed for the cyclomatic complexity defines the number of independent paths through the function. The following calculations were made to determine the cyclomatic complexity of the Cooking Control Function.

Cyclomatic complexity of G,
V(G) = P + 1, where P is the number of predicate nodes (nodes that contain a condition: 2, 3, 4, 5, 6, 7, 8, and 9)
V(G) = 8 + 1
V(G) = 9

An alternate method of calculating the cyclomatic complexity of G is,
V(G) = R, where R is the number of regions on the flow graph
V(G) = 9

9.7.4 Independent Path Table for Cooking Control Function

Number	Path	Test (see 9.7.5)
1	1, 2, 3, 6, 10, 2, 11	Test 1
2	1, 2, 3, 4, 10, 2, 11	Test 2
3	1, 2, 3, 5, 4, 10, 2, 11	Test 3
4	1, 2, 3, 4, 5, 11	Test 4
5	1, 2, 3, 6, 7, 7, 8, 10, 2, 11	Test 5
6	1, 2, 3, 6, 7, 7, 8, 9, 8, 10, 2, 11	Test 6
7	1, 2, 3, 6, 7, 7, 8, 9, 11	Test 7
8	1, 2, 3, 6, 7, 8, 10, 2, 11	Test 8
9	1, 2, 11	Test 9

9.7.5 Test Cases for Each Independent Path of Cooking Control Function.

Test	Step	Action	Expected Result	Actual Result	Pass/Fail	Problem Report Number
Test 1	Step 1	Set cooking time to 00:30 and push Start button.	Oven comes on and the displayed time counts down for 30 seconds to 00:00 at which time the oven shuts off and the beeper sounds			
Test 2	Step 1	Set cooking time to 00:30 and push Start button.	Oven turns on and displayed time begins decrementing from 00:30.			
	Step 2	When time equals 00:15 push Stop/Clear button.	Oven shuts off with 00:15 displayed on LCD screen.			
	Step 3	Immediately push Start button.*	Oven turns back on and time counts down to 00:00 at which time the oven shuts off and the beeper sounds			
Test 3	Step 1	Set cooking time to 00:30 and push Start button.	Oven turns on and displayed time begins decrementing from 00:30.			
	Step 2	When time equals 00:15 push Stop/Clear button.	Oven shuts off with 00:15 displayed on LCD screen.			
	Step 3	Wait approximately 20 seconds..	Oven remains off with displayed time of 00:15			
	Step 4	Push Start button.	Oven turns back on and time counts down to 00:00 at which time the oven			

			shuts off and the beeper sounds			
Test 4	Step 1	Set cooking time to 00:30 and push Start button.	Oven turns on and displayed time begins decrementing from 00:30.			
	Step 2	When time equals 00:15 push Stop/Clear button.	Oven shuts off with 00:15 displayed on LCD screen.			
	Step 3	Wait approximately 20 seconds..	Oven remains off with displayed time of 00:15			
	Step 4	Push Stop/Clear button a second time.	Oven remains off, displayed time changes to 00:00, and the beeper sounds			
Test 5	Step 1	Set cooking time to 01:00 and push Start button.	Oven turns on and displayed time begins decrementing from 01:00.			
	Step 2	When time equals 00:20 open the microwave oven door.	Oven shuts off with 00:20 displayed on LCD screen.			
	Step 3	Wait for approximately 30 seconds. Push Start button to ensure oven stays off with door opened.	Oven remains off with 00:20 showing on the LCD screen.			
	Step 4	Close oven door and immediately push Start button.**	Oven turns back on and time counts down to 00:00 at which time the oven shuts off and the beeper sounds			
Test 6	Step 1	Set cooking time to 01:00 and push Start button.	Oven turns on and displayed time begins decrementing			

			from 01:00.			
	Step 2	When time equals 00:15 open the oven door.	Oven shuts off with 00:15 displayed on LCD screen.			
	Step 3	Wait approximately 20 seconds.	Oven remains off with displayed time of 00:15			
	Step 4	Close the oven door and wait another 20 seconds.	Oven remains off with displayed time of 00:15			
	Step 5	Push the Start button.	Oven turns back on and time counts down to 00:00 at which time the oven shuts off and the beeper sounds			
Test 7	Step 1	Set cooking time to 01:00 and push Start button.	Oven turns on and displayed time begins decrementing from 01:00.			
	Step 2	When time equals 00:15 open the oven door.	Oven shuts off with 00:15 displayed on LCD screen.			
	Step 3	Wait approximately 20 seconds.	Oven remains off with displayed time of 00:15			
	Step 4	Push Stop/Clear button.	Oven remains off, displayed time changes to 00:00, and the beeper sounds			
Test 8	Step 1	Set cooking time to 01:00 and push Start button.	Oven turns on and displayed time begins decrementing from 01:00.			
	Step 2	When time equals 00:15 open the oven door.	Oven shuts off with 00:15 displayed on LCD screen.			
	Step 3	Immediately	Oven turns back			

		close the oven door and push the Start button.***	on and time counts down to 00:00 at which time the oven shuts off and the beeper sounds.			
Test 9	Step 1	Set cooking time to 00:00 and push Start button.	Oven remains off, displayed time is 00:00, and the beeper sounds.			

* Use in-circuit emulator to activate Start signal the frame after the door is opened to prevent the code from branching to node 5 (see flow graph in section 9.7.1).
** Use in-circuit emulator to activate Start signal the frame after the door is opened to prevent the code from branching to node 9 (see flow graph in section 9.7.1).
*** Use in-circuit emulator to branch directly to "T" edges out of nodes 7 and 8 as shown on the flow graph in section 9.7.1.

10. USER'S MANUAL

10.1 TITLE PAGE

Example Project: Microwave Oven User's Manual

TABLE OF CONTENTS

1. INSTALLATION

1.1 GROUNDING INSTRUCTIONS

This appliance must be grounded. Grounding reduces the risk of electric shock. The microwave is equipped with a cord having a grounding wire with a grounding plug. The plug must be plugged into an outlet that is properly installed and grounded.

1.2 PLACEMENT OF THE OVEN

Your microwave oven can be placed easily in your kitchen, family room, or anywhere else in your home. Place the oven on a flat surface such as a kitchen countertop or a specially designed microwave oven cart. Do not place above a gas or electric range. Free air flow around the oven is important.

1.3 UNPACKING THE OVEN

Inspect the oven for damage when unpacking. Check for damage such as dents in the door or inside the heating chamber. Report any dents or breakage to the manufacturer or retailer immediately. **DO NOT ATTEMPT TO USE OVEN IF DAMAGED.** Remove all packing materials from the oven's interior. If the oven has been stored in an extremely cold area, wait a few hours before connecting the power.

1.4 RADIO INTERFERENCE

The operation of the microwave may interfere with TV or radio reception. Interference may be reduced by taking the following steps:

- Clean the oven door and surfaces of the oven;
- Relocate that microwave oven in relation to the TV or radio;
- Move the microwave away from the TV or radio;
- Plug microwave into a different outlet so that microwave oven and radio or TV are on different branch circuits.

2. GETTING STARTED

2.1 Microwave Popcorn: Using Popcorn Preset Function

1) Open the microwave door.
2) Place popcorn in the heating chamber.
3) Press the Popcorn preset key.
4) The timer is automatically set to 3 min. 30 sec.
5) The microwave cooks for the set amount of time.
6) Beeper sounds indicating cooking time has elapsed.
7) Open the door and remove popcorn.

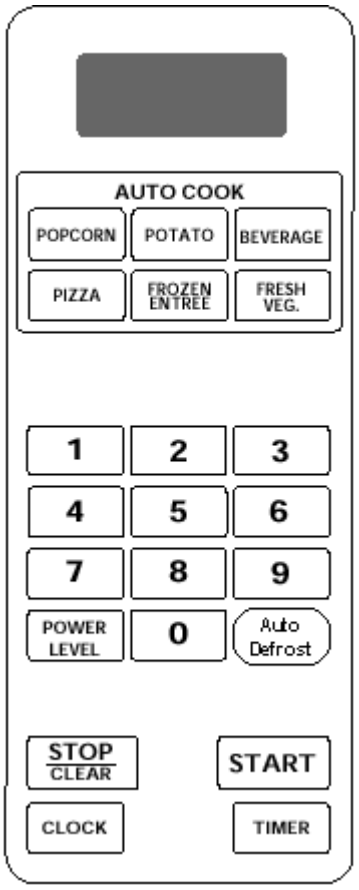
2.2 Microwave Popcorn: Manual Timer Setting

1) Open the microwave door.
2) Place popcorn in the heating chamber.
3) Set the desired cooking time using the numeric keypad.
4) Press the Start key.
5) The microwave cooks for the set amount of time.
6) Beeper sounds indicating cooking time has elapsed.
7) Open the door and remove popcorn.

2.3 Auto Defrost Function: Ground Beef

1) Open the microwave door.
2) Place the item to be defrosted in the heating chamber.
3) Press the Auto Defrost key.
4) The screen will display choices for BEEF, POULTRY, or FISH.
5) Press 1 on the numeric keypad to select BEEF.
6) The screen will then display a prompt for the weight of the item.
7) Enter the weight of the item in ounces using the numeric keypad.
8) Press the Start key.
9) The microwave will automatically set the proper defrost time and power level.
10) The microwave cooks for the set amount of time.
11) Beeper sounds indicating the defrosting time has elapsed.
12) Open the door and remove the item.

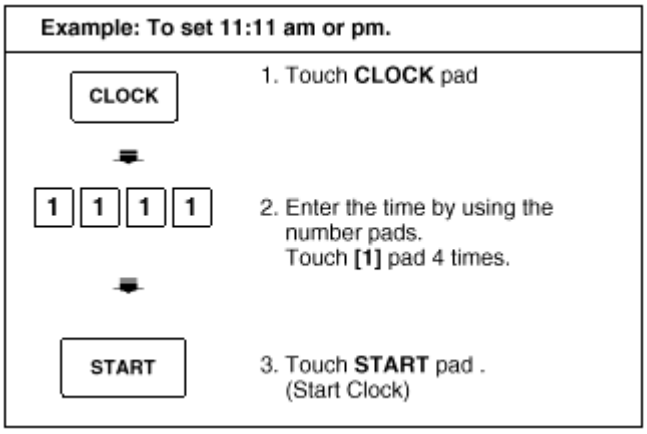
3. USER INTERFACE



4. OPERATIONAL INSTRUCTIONS

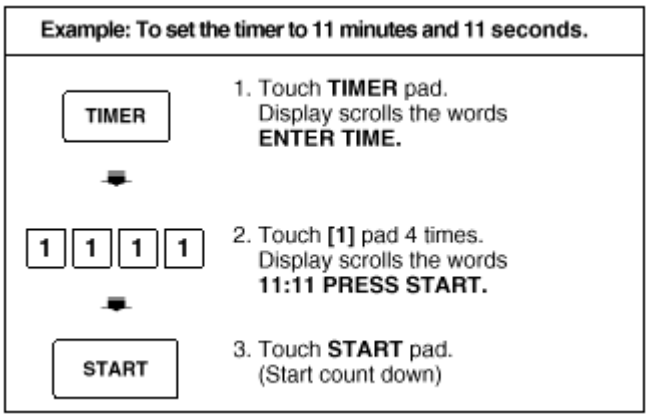
4.1 Setting the Clock

If you are using the microwave oven for the first time or the clock has been reset by a power surge or outage, the words ENTER TIME will appear in the display. Follow the procedure below to set the time.



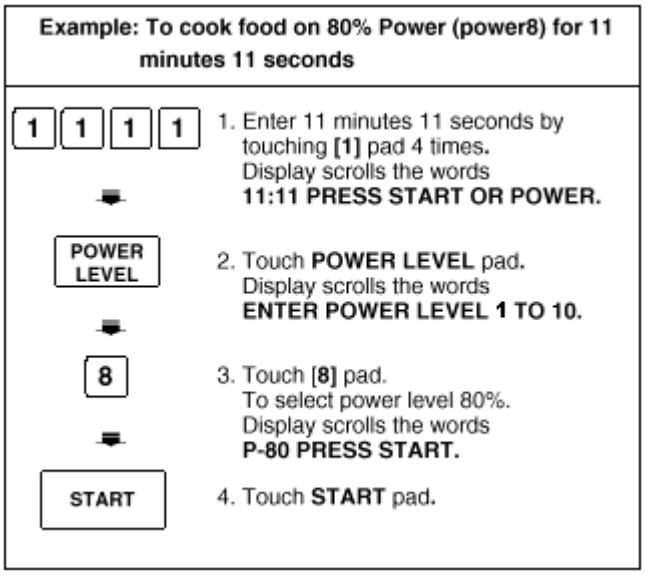
4.2 Timer

The oven's timer function allows the user to set a time that counts down without the oven heating. To set the oven's timer, follow the steps in the diagram below.



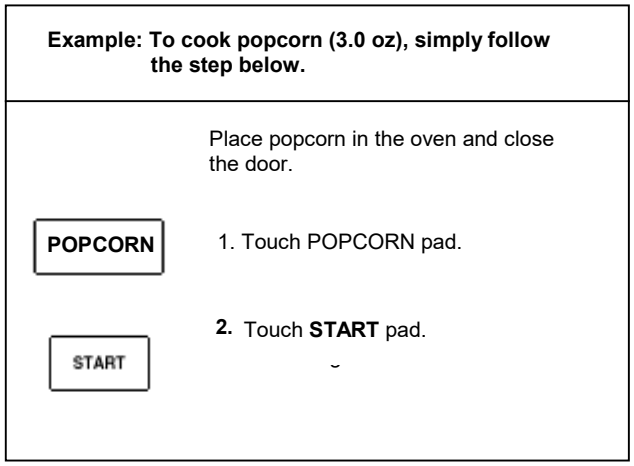
4.3 Timed Cooking

The timed cooking feature allows the user to cook for a desired amount of time at a desired power level. The power level is set by default to HIGH. To cook at another power level for a desired amount of time, follow the example given below.



4.4 Auto Presets

The microwave's interface provides several presets that allow a user to cook certain common food items automatically. To use these presets, follow the example for the Popcorn preset given below.



4.5 Auto Cook Guidelines

4.5.1 POPCORN

1) Follow package instructions.
2) Use only popcorn packages that are made especially for microwave cooking.
3) Heat only one package at a time.

4.5.2 POTATO

1) Place in center of the oven on paper towel.
2) Pierce potato several times with fork.
3) After cooking, let stand for 5 minutes.

4.5.3 BEVERAGE

1) Use microwave-safe cup or mug .
2) Stir after reheating.

4.5.4 PIZZA

1) Place the refrigerated pizza on the plate in the center of the microwave oven.

4.5.5 FROZEN ENTRÉE

1) Follow the package instructions.
2) Example of one frozen entree is lasagna with meat sauce.

4.5.6 FRESH VEGETABLE

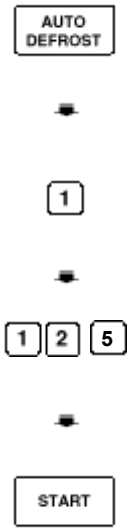
1) Prepare vegetables (wash, cut).
2) Add water as needed.

Note: The auto cook function for fresh vegetables assumes a weight of approximately 10 ounces. Greater or lesser amounts may require more or less cooking time.

4.6 Auto Defrost

The microwave is programmed with three auto defrost settings: MEAT, POULTRY, FISH. The Auto Defrost feature automatically sets the defrost time for the given food item depending on the weight entered by the user. An example of the use of the Auto Defrost feature is given below:

Example: To defrost 12.5oz of ground beef.



1. Touch **AUTO DEFROST** pad.
   Display scrolls the words:
   BEEF PRESS 1
   POULTRY PRESS 2
   FISH PRESS 3.
2. Touch **1** pad.
   Display scrolls the words:
   ENTER WEIGHT.
3. Enter the weight by touching [1], [2], and [5] pads.
   Display scrolls the words:
   12.5oz PRESS START.I
4. Touch **START** pad.
   Defrosting starts.

5. CLEANING INSTRUCTIONS

5.1 OVEN INTERIOR

1) Use a damp cloth to wipe out crumbs and spillovers.
2) It is important to keep the area between door and the front of the heating chamber clean to insure that the door closes properly.
3) Remove grease with a wet cloth, then and dry.
4) Do not use harsh detergent or abrasive cleaners.

5.2 GLASS TRAY

The glass tray can be washed by hand or in the dishwasher.

5.3 CONTROL PANEL

Wipe with a damp cloth followed immediately by a dry cloth.

5.4 OUTSIDE OVEN SURFACE

Clean with damp cloth.

6. TROUBLESHOOTING

If a problem occurs with the unit, use the guide below and try the solution for each problem that occurs.

6.1 OVEN WILL NOT START AT ALL

1) Make sure electrical cord for oven is plugged in.
2) Make sure the door is shut securely.
3) Check operation instructions to make sure current operation is being performed correctly.

6.2 UNEVEN COOKING OR POOR DEFROSTING

1) Make certain the Cooking time/Cooking power level is suitable for the item being heated.
2) Make sure food is turned or stirred properly.

6.3 OVERCOOKED FOODS

1) Make certain the Cooking time/Cooking power level is suitable for the item being heated.

6.4 UNDERCOOKED FOODS

1) Make certain food is defrosted completely.
2) Make certain the Cooking time/Cooking power level is suitable for the item being heated.

INDEX