Modeling and Simulation of a Municipal Solid Waste Management System based on Discrete Event System Specification

Chang-Hyun Lyoo¹, Jinho Jung¹, Changbeom Choi¹ and Eun-Young Kim²

¹Handong Global University, Pohang, Rep. of Korea, {21300259, 21300704, cbchoi}@handong.edu
²Pohang TechnoPark, Pohang, Rep. of Korea, hellosally@daum.net

ABSTRACT
The cleanliness of residential areas is a critical consideration for urban planning. In particular, municipal waste management is a vital service that affects the satisfaction level of residents. Since the satisfaction level may be affected by the amount of garbage accumulated at the time residents put out their garbage, an urban planner may consider residents’ living patterns to develop an efficient waste collection strategy. As the composition of residents in a residential area becomes more complicated, estimating the accumulation of garbage based on the living patterns of residents so as to achieve a high level of residents’ satisfaction with public services becomes an increasingly difficult problem for urban planners. This research focuses on the representation of the temporal behavior patterns of residents in a given area using a discrete event system. Additionally, this research introduces a simulation environment that utilizes the discrete event system specification formalism to simulate the living patterns of residents and to analyze waste generation and the municipal waste management system. The simulation results show that modeling the living patterns of residents based on their occupations may yield a satisfactory representation of the dynamics of a waste management system.

Author Keywords
Agent-based Simulation; Discrete Event System Formalism; Municipal Solid Waste Management System.

ACM Classification Keywords
I.6.5 : SIMULATION AND MODELING (Model Development).

1 INTRODUCTION
Today, almost 55% of the world’s population lives in urban areas, and the urban population is continuing to increase [1]. Among the many issues that arise in an urban environment, the problem of garbage is most closely related to residents’ satisfaction [2]. As the population increases, the total amount of garbage generated in a local area may increase. Consequently, the satisfaction level of residents may decrease when the government does not collect the garbage on time. Moreover, unsatisfied citizens may file civil complaints, leading to an increase in the workload of government employees and higher social costs. To address these issues, urban planners or employees of the local government analyze residential areas to estimate the corresponding amounts of waste generation and ensure the provision of appropriate public services, such as utilizing garbage trucks to collect waste. Despite such efforts, however, it is difficult to estimate waste disposal patterns because the behavior patterns of the residents are more important factors than the geographic or demographic characteristics of the residential area. Notably, residents’ living patterns may differ depending on the cultural and occupational characteristics of different residents; consequently, capturing these living patterns is a difficult problem.

This research proposes a modeling method focusing on human characteristics and behavior patterns to help optimize municipal waste management systems. Since the temporal behaviors of residents may differ based on their living habits and occupations, this research utilizes the discrete event system specification (DEVS) formalism to model residents’ behavior. The DEVS formalism is a set-theoretic framework with modular characteristics [3, 4, 5]. This allows users to develop their own simulation models to serve as components of an urban system or to reuse simulation models built by other practitioners. To maximize the modular capabilities of the DEVS formalism, this research proposes an object-oriented discrete event simulator based on the Python programming language. The simulator captures the common behavior of the residents of an urban area using the DEVS formalism and specializes in analyzing residents based on their occupations. Therefore, practitioners may reuse this DEVS simulation model to simulate the temporal behavior of residents while differentiating residents by their occupations.

2 RELATED WORK AND BACKGROUND
This section introduces and analyzes previous related studies. Additionally, it provides background knowledge to help understand the proposed method and the simulation environment.
2.1 Related Work: Waste Management

Many studies have pursued the optimization of waste management through the application of both analytical solutions and simulation techniques. For example, Dao-Tuan et al. (2017) used integer linear programming and mixed-integer linear programming to optimize the routes of garbage trucks considering emissions control [6].

Some work has been done to optimize the routes of garbage trucks to reduce waste collection costs using agent-based simulations and geographic information system (GIS) techniques. Chalkias et al. (2009) constructed a spatial geodatabase by integrating GIS and municipality databases [7]. They proposed a method of deriving optimal paths for garbage trucks using Dijkstra’s algorithm. Das et al. (2015) solved the waste transportation problem by optimizing waste collection and transportation paths with the aid of the traveling salesman problem to develop a time- and cost-effective waste management system design [8]. Nambiar et al. (2013) compared the static and dynamic vehicle routing problems, with a focus on increasing the capacity utilization of each participating vehicle [9]. Nguyen et al. (2013) proposed a model to optimize the collection and transportation of municipal solid waste in terms of fuel consumption and pollutant emissions [10]. Shi et al. combined hybrid agent-discrete modeling techniques with a Voronoi-based genetic algorithm and a $k$th nearest neighbor search algorithm to build an optimization model [11]. Additionally, they recommended vehicle routing and resource allocation strategies for solid waste management systems. Karadimas et al. (2006) suggested the loose coupling of GIS and simulation software in the domain of municipal waste management [12]. Hua et al. (2016) proposed that applying GIS data and real-time waste data from smart devices to solve routing problems would reduce the travel distance of the collection vehicles [13]. System dynamics approaches have also been used in forecasting waste generation and simulating an efficient waste management system [14, 15, 16]. Dyson et al. (2005) predicted waste generation using 5 different models [15]. Guariso et al. (2009) predicted the limit on the performance of a waste transfer station in Australia [14]. Wang developed a waste prediction and response model based on system dynamics [16].

However, to the best of the authors’ knowledge, there has been little discussion of the modeling of human behavioral patterns or cultural characteristics in the context of waste management. This study proposes a modeling method to capture the behaviors of residents using the DEVS formalism.

2.2 Background: Discrete Event System Specification (DEVS)

The DEVS formalism is a set-theoretic formalism developed for specifying discrete event systems. The DEVS formalism comprises two types of models to represent discrete event systems: atomic models and coupled models. There are two ways to build a model using the DEVS formalism. First, the user may specify the behavior of some indivisible component of the discrete event system using an atomic model. On the other hand, the user may compose a system represented by a coupled model by assembling multiple atomic models or other coupled models.

In this research, an atomic model specification is adopted to represent the behavior of the simulated entities in an urban simulation to capture the dynamics of the waste management system and garbage disposal patterns. Since the proposed urban simulation environment manages various simulation models and the connections between them, this simulation environment is a coupled model. However, the user need not consider the specification of the coupled model because the simulation environment manages the insertion of and connections between different entities in the urban simulation. The atomic model specification is defined as follows:

$$AM = < X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta >$$

where

- $X$: a set of external input event types;
- $Y$: an output set;
- $S$: a sequential state set;
- $\delta_{ext}$: $Q \times X \rightarrow S$, an external transition function, where $Q$ is the total state set of $M = \{(s,e)|s \in S \ and \ 0 \leq e \leq ta(s)\}$;
- $\delta_{int}$: $S \rightarrow S$, an internal transition function;
- $\lambda$: $S \rightarrow Y$, an output function;
- $ta$: $S \rightarrow R_{0, \infty}^+$, a time advance function,

where $R_{0, \infty}^+$ is the set of non-negative real numbers up to $\infty$.

An atomic model $AM$ is a model that is affected by external input events $X$ and generates output events $Y$ based on the model state. The state set $S$ represents the unique description of the model. The internal transition function $\delta_{int}$ and the external transition function $\delta_{ext}$ are used to compute the next state of the model. If an external event arrives after an elapsed time $e$ that is less than or equal to the value $ta(s)$ specified by the time advance function $ta$, then the new state $s'$ is computed using the external transition function $\delta_{ext}$. Subsequently, a new $ta(s')$ is computed, and the elapsed time $e$ is set to zero. Otherwise, when an internal event arrives at $ta(s)$, the new state $s'$ is computed using the internal transition function $\delta_{int}$. In the case of an internal event, the output specified by the output function $\lambda$ is produced based on the state $s$, which means that the output function is processed before the internal transition function. Then, as before, a new $ta(s')$ is computed, and the elapsed time $e$ is set to zero.

![Figure 1. Example of an Atomic Model](image-url)

Figure 1 shows a diagram of an atomic model. The label starting with a question mark denotes the input port, and the
label starting with an exclamation mark denotes the output port. The ellipses represent the states of the model, and the ellipse indicated with an arrow denotes the initial state of the model. Each state is associated with a state name and a positive real number that represents the occupancy time. When the elapsed time is equal to the occupancy time, the simulation algorithm will sequentially trigger the output function and the internal transition function.

The proposed simulator partially relies on the notation of the DEVS formalism and its simulation algorithm. Listing 1 shows the Python code for the example atomic model. As shown in Listing 1, the proposed simulator provides a programming interface that has a one-to-one correspondence with the atomic model specification of the DEVS formalism.

```python
class AtomicModel(BehaviorModelExecutor):
    def __init__(self, instance_time, destruct_time, name, engine_name):
        BehaviorModelExecutor.__init__(self, instance_time, destruct_time, name, engine_name)

        self.init_state("State1")
        self.insert_state("State1", Infinite)
        self.insert_state("State2", 0)
        self.insert_input_port("in_port")
        self.insert_output_port("out_port")

    def ext_trans(self, port, msg):
        if port == "in_port":
            self._cur_state = "State2"
        elif port == "out_port":
            return TimeStruct(21, 00, Statistic(0, 0, 1))

    def output(self):
        name = self.get_name()
        msg = SysMessage(name, "out_port")
        return msg

    def int_trans(self):
        if self._cur_state == "State2":
            self._cur_state = "State1"

Listing 1. Example Code for an Atomic Model
```

3 PROPOSED METHOD AND ENVIRONMENTS

This section introduces the modeling results for the simulated entities of the municipal solid waste management system and its simulation environment. In this research, we propose two methods for modeling a municipal waste management system. First, we separate the domain-related part and the simulation-related part. Second, we adopt an object-oriented concept to allow practitioners to select entities to be simulated and synthesize the desired simulation based on the target situation. In this way, experts with various levels of understanding may participate in simulation development and experimentation.

To capture the dynamic behavior of the residents, we classify the residents into different types and model them using the DEVS formalism. The parameters required for simulation are modeled by defining a corresponding class in the Python programming language. Additionally, simulated entities that are modeled using an atomic model inherit the properties of the base class for discrete event simulation. Each class instance of each resident type and the discrete event simulation model are synthesized during the initialization phase of the simulation process.

As an example, Listing 2 shows the code for the resident-type class for a student. The code returns information on the resident’s behavior, such as the wake-up time, sleep time, and amount of waste generated. Each piece of information is based on a survey titled “What time did you wake up today?” conducted by Gallup Korea in 2013 [17].

```python
class Student(HumanType):
    def __init__(self, _id):
        HumanType.__init__(self, _id)

    def get_type(self):
        return "Student"

    def get_sleep(self):
        return TimeStruct(7, 58, Statistic(0, 0, 1))

    def get_in(self):
        return self.get_wakeup() + 1

    def get_trash(self):
        return 0.3

    def get_satisfaction_func(self, trash):
        if trash >= 0.8:
            return 10
        elif trash <= 0:
            return -10
        else:
            return 0.3

Listing 2. Example Code for the Resident-Type Class for a Student
```

Figure 2 shows the overall framework of the proposed simulation environment. The solid rectangles represent real models of the behavior of simulated entities instantiated in the simulation environment. Each real model implements the atomic behavior of the corresponding simulated entity. On this basis, an urban planner may compose various simulation models of interest. For example, some less sensitive residents may not file complaints to the government. Accordingly, the urban planner may use only the basic resident model to model such a less sensitive resident rather than modifying the parameters of the simulated entity. The dotted rectangles represent conceptual models. Each conceptual model may comprise several real models. When a practitioner inserts real models into the simulation environment, the simulation environment connects the ports of the newly introduced entities to those of other entities.

Figure 3 shows a diagram of the resident model used to model a resident of the residential area. The resident model can calculate the departure time of a resident based on the wake-up time of the corresponding resident-type instance, as shown...
in Listing 2. The practitioner may reuse this resident model for different resident-type object instances. Notably, this research assumes a normal distribution for the departure time to support stochastic simulation. During the simulation, the resident model may generate a trash event with a given amount of waste generation from a particular resident-type instance and send it to the corresponding instances of the family model and check model.

Figure 3 shows a diagram of the check model. This is the model used to measure the satisfaction level of a resident in accordance with the capacity of the garbage can. Since most residents do not check the garbage during the daytime, the check model checks the garbage can at the departure time of the resident. Therefore, the check model checks the garbage can periodically upon departure events. Since the resident model generates garbage events based on the departure time, the check model utilizes these events to handle departure events. As a result, the level of satisfaction of a resident with the cleanliness of the garbage collection site is measured every departure time. When the check model receives a request message from the resident model, a check message is sent to the garbage can model to check the status of the garbage can. The check model then passes the value received from the garbage can through the check port to the resident-type instance. Each resident-type class has a function that calculates the satisfaction level based on the accumulation ratio of the garbage can. After checking the satisfaction level, the check model may trigger a claim message to the government model through the gov-report port if the satisfaction level is below a set threshold.

Figure 5 shows a diagram of the garbage can model. The garbage can model models the temporary garbage collection area for each building. Each instance of the garbage can model may have a different amount of free space during the
simulation. If residents dispose of a large amount of garbage in the garbage can or a garbage truck does not collect sufficient garbage from the garbage can, some garbage may remain in the garbage can. The remaining garbage may affect the satisfaction level of a resident. Therefore, the practitioner may observe changes in residents’ satisfaction depending on the garbage can capacity and estimate a suitable capacity according to the residents. When the garbage can model receives an event from the trash_from_family port, the garbage can is filled with an amount of garbage equal to the given value. When the garbage can model receives an event from the check model, the garbage can model calculates the accumulation ratio of the garbage can and sends the result back to the check model. When the garbage can model receives a message from the garbage truck model, the accumulated amount of garbage is sent to the garbage truck model.

Figure 6 shows a diagram of the garbage truck model. This model is designed to retrieve garbage from buildings at regular intervals. The parameters of the garbage truck model are the visit order of the buildings, the collection time for each building, the capacity of the garbage truck, and the collection cycle. Each parameter can be adjusted to create an optimized waste management system. When the garbage truck model visits a building, garbage is taken from the garbage can and added to the truck. The truck has a fixed storage size, which the amount of garbage that the truck is carrying cannot exceed. Therefore, the garbage truck may not collect garbage from a garbage can when the truck does not have sufficient storage space available. The garbage truck model sends a message to receive the accumulated garbage from a garbage can. This process is repeated until the truck has visited all buildings. After the truck has visited all buildings, the amount of garbage in its storage space is set to zero, and another round of garbage collection is scheduled.

Figure 7 shows a diagram of the government model. The government model collects complaints from residents. To assist the practitioner, the model records the complaint time and satisfaction level of each resident.

4 CASE STUDY

This section illustrates a case study of the proposed method and simulation environment. Based on the population composition of an administrative district in Pohang City, Republic of Korea, experiments were conducted by modeling the living patterns of construction workers, students, and homemakers, who represent the majority of the occupants of the administrative districts. The living pattern for each occupation was compiled based on the analysis report for a public opinion survey targeting the national population [17].

The following assumptions were adopted when modeling the residential area and the waste management system:
- Each building has a garbage can to hold garbage temporarily.
- Each resident departs on his or her own schedule.
- Each resident takes out the garbage every day when he or she departs.
- The satisfaction level of a resident may change at the moment when the resident takes out the garbage.
- A resident may complain to the government when the garbage accumulation is above a certain level.

4.1 Experimental Design

Because the actual operations of garbage trucks may be different for each garbage disposal company, in this research, an abstract model was built for the garbage collection strategy. The garbage truck collects trash from the garbage can of each building based on the current capacity of the truck. Additionally, single-person households were assumed to analyze the simulation results with regard to the following patterns: 1) a fixed sequence, 2) a palindrome sequence, and 3) a random sequence. A fixed sequence is a sequence in which the garbage truck visits the buildings in a predefined order. For example, if there are three buildings, A, B, and C, the garbage truck may visit the buildings in alphabetical order. Accordingly, the sequence of the schedule would be A → B → C → A → B → C → ... A palindrome sequence is a sequence in which the order is reversed after the completion of each cycle of collection. In the palindrome sequence for the example given above, the garbage truck would visit the last building first in the next cycle, and the building visited first in the previous cycle would be visited last, as follows: A → B → C → C → B → A → ... Finally, a random sequence is a sequence in which every building is visited in a random order.

The experiments were conducted using a system with a 4.20 GHz Intel(R) Core(TM)i7-7700K CPU with 4 physical cores and 64 GB of RAM running 64-bit Windows 10 Pro, build 18362. Ubuntu 18.04.4 LTS (x86_64 GNU/Linux 4.19.76-linuskit) and Python 3.6.9 were mounted using Docker. Each experiment was conducted 30 times to measure the average number of complaints from residents concerning each waste collection strategy: fixed, palindrome, and random. On average, each simulation required 333 seconds to simulate 31 days.
Table 1. Description of Simulation Parameters

<table>
<thead>
<tr>
<th></th>
<th>Construction Worker</th>
<th>Student</th>
<th>Homemaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Departure Time</td>
<td>6:22:00</td>
<td>7:58:00</td>
<td>13:00:00</td>
</tr>
<tr>
<td>Amount of Waste Generated</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>Satisfaction Function</td>
<td>[ Sat(x) = \begin{cases} +10 &amp; \text{if } x &lt; 0.8, \ 20 &amp; \text{if } x = 0, \ -10 &amp; \text{if } x &gt; 0.8 \end{cases} ]</td>
<td>[ Sat(x) = \begin{cases} +10 &amp; \text{if } x &lt; 0.8, \ 20 &amp; \text{if } x = 0, \ -10 &amp; \text{if } x &gt; 0.8 \end{cases} ]</td>
<td>[ Sat(x) = \begin{cases} +10 &amp; \text{if } x &lt; 0.8, \ 20 &amp; \text{if } x = 0, \ -10 &amp; \text{if } x &gt; 0.8 \end{cases} ]</td>
</tr>
</tbody>
</table>

Table 1 shows the simulation parameters deduced from the assumptions. *Departure Time* represents the interval at which a resident leaves his or her home. We modified the *Departure Time* based on the average wake-up time data from the behavior survey [17]. *Amount of Waste Generated* represents the amount of waste generated between departure times. To ensure a realistic simulation, the amount of waste generated by residents was drawn from a distribution corresponding to the average waste generation per capita [18]. The *Satisfaction Function* represents a resident’s satisfaction level depending on the level of garbage accumulation. This function was defined heuristically based on the experiences of several students.

### 4.2 Simulations and Lessons Learned

Figure 8 shows the simulated numbers of complaints filed by a population with an equal composition of students, homemakers, and construction workers (33:33:33) served by a garbage truck following a fixed sequence. At 7:00, the number of complaints from students was the lowest, and the number of complaints from homemakers was the highest. As the collection time was delayed from 7:00 to 12:00, the number of complaints from students continuously increased, and the number of complaints from homemakers decreased. As the collection time was further shifted from 13:00 to 14:00, the number of complaints from students decreased, and the number of complaints from homemakers increased. In the case of the construction workers, the number of complaints increased gradually from 7:00 to reach a maximum at 9:00 and then continuously decreased until 14:00. In terms of total hourly complaints, the lowest number of complaints occurred at 13:00, and the highest number of complaints occurred at 9:00. The average number of complaints across all times was 280.87.

Figure 9 shows the simulated numbers of complaints filed by a population with an equal composition of students, homemakers, and construction workers served by a garbage truck following a palindrome sequence. The number of complaints from students continuously increased as the collection time was delayed from 7:00 to 12:00, while the number of complaints from homemakers decreased. This trend reversed at 12:00, with the number of complaints from students decreasing from 13:00 to 14:00, while the number of complaints from homemakers increased. In the case of the construction workers, the number of complaints increased from 7:00 to 9:00 and decreased beginning at 12:00. Overall, the lowest number of complaints was at 7:00, and the highest number of complaints was at 14:00. The average number of complaints across all times was 332.52.

Figure 10 shows the simulated numbers of complaints filed by a population with an equal composition of students, homemakers, and construction workers served by a garbage truck following a random sequence.
following a random sequence. The number of complaints from homemakers was the highest when the garbage collection time was at 7:00, while the students filed the highest number of complaints at 12:00. Construction workers made the most complaints at 9:00. Across all times, the total number of complaints was lowest when garbage was collected at 9:00 and the highest at 12:00. The average number of complaints across all times was 353.14. Among the three garbage collection strategies, the fixed sequence resulted in an average number of civil complaints that was 18 to 25% lower than those in the cases of the palindrome and random sequences in these experiments. The actual waste collection company operating in the study area follows a fixed collection schedule, that is, a fixed sequence, which results in the lowest average number of complaints.

Figure 11 shows the simulated numbers of complaints filed by a population with a composition of 18:33:48 (students:homers:construction workers) served by a garbage truck following a fixed sequence. At 7:00, the number of complaints from students was the lowest, and the number of complaints from homemakers was the highest. At 12:00, students and construction workers filed the most complaints. Overall, the total number of complaints was the highest at 12:00 and the lowest at 13:00. The patterns of increase and decrease in the numbers of complaints from residents with each occupation were similar to those in the case of equal composition rates. The average number of complaints across all times was 280.69, almost the same as in the equal-composition simulation. From these experiments, it was found that the results vary depending on the population composition. However, with fixed-sequence collection, the average number of complaints is the lowest for collection at 13:00, even for different population compositions. It can be concluded that changing the garbage collection time to 13:00 should reduce the average number of complaints.

**5 CONCLUSION**

As the complexity of urban areas increases, estimating waste generation and providing alternative solutions for waste collection are essential for urban planners and employees of the local government to increase the satisfaction level of residents. In this research, the living patterns of residents with different occupations were modeled to estimate waste generation and collection to compute the residents’ approximate satisfaction level. In the proposed modeling approach, the simulation environment was modeled by considering residents, buildings, garbage trucks and different operation strategies based on the DEVS formalism. In this environment, various assumptions and operation strategies were established and tested.

The experimental results showed that the number of complaints may vary depending on the population composition; however, the total number of complaints was the lowest when garbage was collected at 13:00, independent of the population composition, for the case of fixed-sequence collection in the given experimental environment. This study has demonstrated that it is possible to perform modeling based on the population composition of a residential area in an actual city and thus determine the optimal collection time on the basis of the existing collection pattern. Alternatively, the proposed modeling approach and simulation environment can be used in combination with alternative derivation methods to derive the optimal collection pattern. In future research, an experiment can be conducted to investigate family-unit waste disposal patterns, ranging from single-person households to multi-person households. Additionally, an experiment to validate the effectiveness of installing smart garbage cans can be performed by adopting dynamic vehicle routing in the collection schedule. Implementing a graphical user interface to increase the usability of the model for urban planners or local government employees can also be considered.

**REFERENCES**


