

## 도시가스 누출에 따른 스마트 홈 안전 플랫폼 프로그램 구축

지현민 · 이우귀연 · †오정석

한국가스안전공사

(2019년 10월 29일 접수, 2019년 12월 20일 수정, 2019년 12월 21일 채택)

## Smart Home Safety Platform Program for City Gas Leakage

HyunMin Ji · Ugwiyeon Lee · †Jeongseok Oh

Korea Gas Safety Corporation

(Received October 29, 2019; Revised December 20, 2019; Accepted December 21, 2019)

### 요 약

본 연구에서는 우리나라 가정에서 가장 많이 쓰이고 있는 LNG를 대상이며 수요가 증가함에 따라 가스사고는 늘어나고 있는 실정이다. 그 중 사용자취급부주의로 인한 사고가 가장 많은데 이를 효과적으로 미연에 방지 하고자 시나리오를 구축하여 정보제공을 하고자한다. 가정에서의 가스누출은 밀폐 된 공간이며 이에 따른 적합한 gaussian model을 적용하였다. 4차 산업 혁명으로 스마트 홈 제어 기술이 발전함에 따라 가스누출사고를 예측하고 방지 하고자 프로그램으로 정보제공을 할 수 있는 토대를 만들어 스마트 홈 안전 플랫폼에 적용하고자 한다.

**Abstract** - In this study, city gas is the most commonly used in homes, and gas accidents are increasing as demand increases. Among them, accidents due to user carelessness are the most and we try to provide information by establishing scenarios to effectively prevent this. The gas leak in the home is an enclosed space, and the appropriate Gaussian model is applied. As smart home control technology is developed by the 4th industrial revolution, it will make a foundation to provide information through program to predict and prevent gas leakage accidents and apply it to smart home safety platform.

**Key words** : Gas Leak, LNG, Gaussain, Smart home

### I. 서 론

도시가스는 우리나라 가정에서 가장 많이 쓰이고 있는 가스이며, 가스안전 공사의 2018년도 가스통계에 따르면 국내 도시가스 수요는 계속 증가 중에 있다 [1]. (Table 1.) 수요가 많은 만큼 도시가스 사고 또한 빈번하게 일어나고 있는 실정이다. 최근 10년간 가스별 사고로 도시가스가 252건(19.2%)로 높은 비율을 차지하고 있다. (Table 2.) 이 중 원인별 사고는 사용자취급부주의가 가장 높은 비중을 차지하고 있으며, 도시가스 사고는 가정에서 인명, 재산피해가 발생하므로 도시가스 사용에 관한 근본적인 사전 안전성강화가 요구되고 있다.

가정에서의 도시가스 누출은 대부분 밀폐된 공간에서 발생되기 때문에 심각한 피해를 발생시키며, 가스 누출 발생 시 환기 조치가 이루어지면 화재나 폭발 사고를 미연에 방지할 수 있을 것이다.

최근 IoT 기술의 발전으로 사전에 가스 누출을 감지하여 자동으로 누출을 차단됨과 동시에 환기시스템을 작동시켜 사전위험으로부터 안전하도록 시스템을 구축하고 있다.

이에 따라 본 연구는 도시가스 시설 내 누출이 발생하여 환기 시스템이 작동하여 폭발하한계(LEL)이하가 되어 안정한 상태가 되는 시간을 산출하여 스마트 홈 안전플랫폼에 적용하고자 한다.

†Corresponding author:jsoh90@gmail.com

Copyright © 2019 by The Korean Institute of Gas

**Table 1.** Gas demand per year for the last five years as of 2018, 2018 Gas Statistics of Korea Gas Safety Corporation[2]

구분	총가구수	LPG 사용 수요가 수			도시가스 사용 수요가 수					가스사용 수요가수 (계)
		주택용	영업용	소계	단독주택	공동주택	산업용	상업+ 업무용	소계	
2013	21,919,518	4,832,244	492,841	5,325,085	4,865,546	10,758,798	14,789	736,105	16,375,238	21,700,323
2014	22,170,727	4,595,065	472,740	5,067,805	5,122,114	11,006,915	15,021	737,165	16,881,215	21,949,020
2015	22,422,161	4,374,992	461,688	4,836,680	5,262,108	11,327,741	15,024	758,652	17,363,525	22,200,205
2016	22,767,734	4,109,326	452,474	4,561,800	5,433,777	11,750,906	14,996	779,625	17,979,304	22,541,104
2017	23,134,508	3,886,841	451,435	4,338,276	5,592,589	12,153,421	15,081	804,980	18,566,071	22,904,347
2018	23,551,089	3,737,394	447,260	4,184,654	5,730,166	12,575,387	14,861	812,047	19,132,461	23,317,116

**Table 2.** Cause of accident by year for last 5 years as of 2017, 2017 Gas Year Book of Korea Gas Safety Corporation[3]

구분	2013년	2014년	2015년	2016년	2017년	계	점유율(%)
계	121	120	118	122	121	602	100
사용자취급부주의	41	41	41	38	31	192	33.3
공급자취급부주의	12	9	5	3	3	32	6.4
타공사	2	6	4	8	7	27	3.8
시설미비	23	17	22	26	29	117	18.8
제품노후(고장)	4	14	19	12	18	67	8.6
*기타	18	17	11	23	24	96	16.3
고의사고	21	16	16	12	9	74	12.7

\*기타(24) : 교통사고(8), 작업부주의(7), 시공 작업(3), 과열화재(2), 원인미상(1), 용기취급부주의(1), 기타(2)

## II. 개발환경

### 2.1. 개발환경

#### 2.1.1 JAVA

자바스크립트는 본래 웹 페이지의 사용자 인터랙션을 담당하는 언어이다[4]. 최근에는 자바스크립트는 웹 어플리케이션으로 확장되어 사용되고 있고 다양한 기능의 어플리케이션을 브라우저에서 실행할 수 있도록 구현이 가능하며 HTML, CSS, 자바스크립트 기반의 웹 언어로 작성되어 이식성이 좋고 가장 많이 사용되고 있다[5]. 자바의 가장 큰 특징은 플랫폼의 독립성을 갖고 있다는 점이다. 다른 프로그램을 작성할 경우 운영체제나 CPU에 따라 알맞게 다시 수정해야 하는 작업(포팅)을 해야 했다. 하지만 자바프로 그

램의 경우 한번 작성된 프로그램은 운영체제나 CPU에 상관없이 동작이 가능하다는 점이다. 향후 프로그램을 발전시켜서 실시간으로 어디서나 사용할 수 있는 환경을 구성하기 최적인 프로그래밍 언어라고 할 수 있다

자바의 최대 강점은 웹 및 안드로이드에 가장 최적화 되어있기 때문에 가스누출 및 사고가 발생하였을 때 핸드폰으로 실시간으로 확인이 가능한 장점이 있어 스마트 홈 안전플랫폼을 적용하기에 가장 적합한 언어이기에 채택이 되었다.

Java 8 SE(Fig 1.)의 대표적인 구조로는 JDK(Java Development Kit), JRE(Java Runtime Environment), API(Application Programming Interface)가 있다.

2.1.2 Eclipse

Java 개발 도구에는 여러 가지가 있다. 그 중에서 이클립스[7]는 오픈 소스 플랫폼으로 인터넷에서 쉽게 구할 수 있을 뿐 아니라 사용에도 제한이 없기 때문에 가장 널리 쓰이고 있다.

이클립스란 자바 기반의 개방적이고 확장성을 지닌 통합 개발 환경이다. 이클립스 플랫폼은 소프트웨어를 개발하고, 통합 소프트웨어 개발 툴을 만들고 실행하는 기반을 제공한다.

이클립스의 구조(Fig 2.)는 플랫폼 자체의 모든 측면을 포함하여 모든 기능은 플러그인을 통해 제공되며, 이클립스는 통합개발환경(IDE) 제품들에 대한 핵심 요소로서, 플랫폼 런타임(Platform Runtime), SWT, JFace, 워크스페이스(Workspace/Resources), 워크벤치(Workbench) IDE UI 등의 컴포넌트로 이루어져 있다.

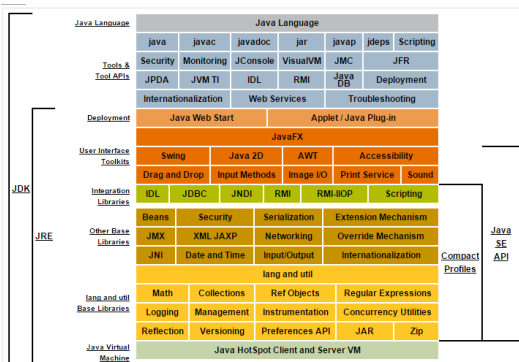


Fig. 1. Java 8 SE Structure.[6]

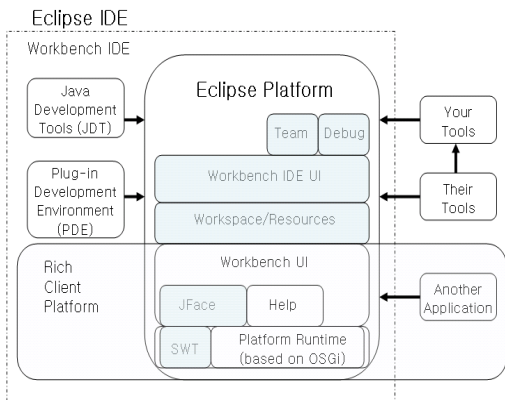


Fig. 2. Eclipse platform Structure[8]

III. 관련 이론

2.2. 관련이론

2.2.1 모델설정

2.2.1. 확산모델

증기의 대기 확산에 대한 예상은 정량적 위험분석(CPQRA) 결과 예측에 핵심적이다. 2가지 종류의 증기운의 움직임과 3가지 서로 다른 배출 시간 모드를 정의 (Table 3.)할 수 있다. 잘 알려진 가우스 모델은 (Gaussian model)은 주어진 풍속, 풍향에서 누출된 부유 가스의 움직임에 대해 설명한다. 따라서 누출로부터 바람이 부는 방향으로 떨어진 거리의 증기운의 움직임으로 부유 연속누출 증기운(Plume) 과 순간 누출 증기운(Puff)로 나눌 수 있다.

전형적인 부유 증기 운은 다양한 대기 조건과 관련하여 다양한 형태를 가지고 있으며 대기가 안정되어 있으면 혼합이 최소화되고, 불안정하면 최대이며 안정된 대기상태에 따라서 A~F등급(Table 4.)으로 나뉘게 되고 F로 갈수록 안정된 상태라고 볼 수 있다. 대기상태의 계산은 식(1)에 따라서 식(2)를 따르게 된다.

$$(C)(x, y, z, t) = \frac{G^*}{(2\pi)^{3/2} \delta_x \delta_y \delta_z} \exp\left[-\frac{1}{2} \left(\frac{y}{\delta_y}\right)^2\right] \quad (1)$$

$$* \exp\left[-\frac{1}{2} \left(\frac{z-H}{\delta_z}\right)^2\right] + \exp\left[-\frac{1}{2} \left(\frac{z+H}{\delta_z}\right)^2\right]$$

Table 3. Steam cloud and leak time types

증기운의 움직임	누출 지속 기간
중성 부유 가스 (양성) 부유가스 무거운 (또는 음성 부유)가스	순간적 누출(증기운) 연속 누출(증기운) 시간에 따라 변하는 지속성

Table 4. Recommended Equations for Pasquill-Gifford Dispersion Coefficients for puff Dispersion [10]

Stability class	$\delta_y$ or $\sigma_x$	$\delta_z$
A	0.18x0.92	0.60x0.75
B	0.14x0.92	0.53x0.73
C	0.10x0.92	0.34x0.71
D	0.06x0.92	0.15x0.70
E	0.04x0.92	0.10x0.65
F	0.02x0.89	0.05x0.61

$$\exp\left[\frac{1}{2}\left(\frac{x-ut}{\delta_x}\right)^2\right] \quad (2)$$

2.2.2. 구멍을 통한 증기의 흐름

기체와 증기의 배출은 조절방출(throttling)과 자유 팽창(free expansion)방출로 분류된다. 조절방출의 예는 마찰손실이 큰, 작은 틈을 통한 방출이며 물리적 구조에 대한 자세한 정보를 필요로 하기 때문에 고려하지 않는다. 자유팽창 방출의 경우 대부분의 압력에너지는 운동에너지로 전환되며 등 엔트로피과정(isentropic behavior)의 타당성을 갖고 누출공의 지름만을 필요로 한다. 이를 등엔트로피팽창에 대하여 이상기체법칙을 식(3)으로 사용[11]하여 질량유속으로 나타내면 다음과 같은 식(4)의 결과식이 얻어진다. 이를 가스별 몰 질량(molecular weight), 열용량비(Heat capacity ratio)를 대입한다.

$$\rho = \rho_0 \left(\frac{P}{P_0}\right)^{1/\gamma} \quad (3)$$

$$Qm = C_D A P_1 \sqrt{\frac{2g_c M}{R_g T_1} \frac{k}{k-1} \left[ \left(\frac{P_2}{P_1}\right)^{2/k} - \left(\frac{P_2}{P_1}\right)^{(k-1)/k} \right]} \quad (4)$$

IV. 시나리오 설정

4.1. 도시가스 누출 공간 모델

도시가스 누출은 대부분의 가정 내 밀폐된 공간에서 이루어지므로 건물 내에 바람이 없는 모델(Fig. 3.)에서 가스가 누출된 상황이다. 이 후 가스검지기가 가스 누출을 감지 한 후 밸브가 차단되어 더 이상은 가스가 누출이 되지 않는 상황에서 순간 누출 증기운(Puff) 공식을 적용하여, 이 후 환풍기를 통해 바람이 외부로 빠져나가면서 시간에 따른 확산농도를 예측 한 모델이다.

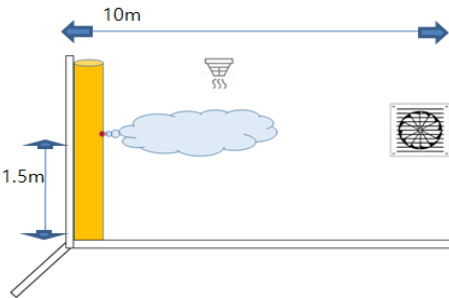


Fig. 3. Indoor City Gas Leakage Model without Wind.

모델에 대한 가스누출(높이, 직경), 환풍구 거리, 검지시간, 위치는 공간에 따라 변경이 가능하며 실내에서 일반적인 대기안정도D등급을 선택하고 사람의 높이인 1.5m에서의 누출을 가정하여 환풍구가 누출 원으로부터 10m거리에 위치해 있을 때를 가정하여 시뮬레이션 하고자한다.

4.2. 도시가스 누출 프로그램 설정

도시가스 주성분인 메탄을 적용하여 초기 값을 (Table 5.)와 같이 설정하였다.

이렇듯 도시가스 누출에 따른 농도변화 프로그램은 누출모델과 가스물성을 적용하여 Java언어로 코딩하여 이클립스를 통해 (Fig 4.)와 같이 구현 하였다.

Table 5. City gas properties and set values of the program

시뮬레이션 설정 값			
비중	0.61	폭발하한계	50000ppm
주변압력	1.01bar	배관압력	1.03bar
물질량	17.4	실내온도	25°C
열용량율비	1.194	누출구 직경	0.01m
대기안정도	D	누출구 높이	1.5m
흡입환기량	0.3m/s	환풍기위치	10m

```

1 package safetyPlatform;
2
3 import java.io.IOException;
4
5
6 /**
7  * Servlet implementation class asdf
8  */
9
10 public class Main extends HttpServlet {
11     private static final long serialVersionUID = 1L;
12
13     /**
14      * GET method for handling requests
15      */
16     public void doGet(HttpServletRequest request,
17         HttpServletResponse response)
18         throws ServletException, IOException {
19         // Auto-generated constructor stub
20     }
21
22     /**
23      * POST method for handling requests
24      */
25     public void doPost(HttpServletRequest request,
26         HttpServletResponse response)
27         throws ServletException, IOException {
28         // Auto-generated constructor stub
29     }
30
31     /**
32      * Initialize the variables
33      */
34     public void init() throws ServletException {
35         // the mass discharge coefficient [dimensionless]
36         double CDeff = 1.0;
37         // Ambient Pressure [bar]
38         double outPressure = 1.01;
39         // Physical Gas Average Mol[1-10] Helium: 0.0 Nitrogen: 0.1 Argon: 0.1
40         double heatCapacityRatio = 1.194; // Natural Gas heat_capacity_ratio [dimensionless]
41         double LEL = 50000; // LEL: 5% 인가 50000ppm
42
43         // 초기 온도 [K]
44         double temperature = 298; // 77°F를 위한 상온을 일반적으로 298K 사용 [bar]
45         // 실내 온도 [K]
46         double temp = 298;
47
48         // 초기 압력 [bar]
49         double inPressure = 1.03;
50         // 환풍기 위치 [m]
51         double inletsDist = 0.0;
52         // 환풍기 높이 [m]
53         double inletsHeight = 1.5;
54
55         // 초기 방출 속도 [m/s]
56         double velocity = 0.3;
57         // 초기 방출 위치 [m]
58         double ventDist = 5;
59         // 방출기 위치 [m]
60         double ventHeight = 1.5;
61         // 방출기 위치 [m]
62         double ventDist = 5;
63         // 방출기 위치 [m]
64         double ventHeight = 1.5;
65         // 방출기 위치 [m]
66         double ventDist = 5;
67         // 방출기 위치 [m]
68         double ventHeight = 1.5;
69         // 방출기 위치 [m]
70         double ventDist = 5;
71         // 방출기 위치 [m]
72         double ventHeight = 1.5;
73         // 방출기 위치 [m]
74         double ventDist = 5;
75         // 방출기 위치 [m]
76         double ventHeight = 1.5;
77         // 방출기 위치 [m]
78         double ventDist = 5;
79         // 방출기 위치 [m]
80         double ventHeight = 1.5;
81         // 방출기 위치 [m]
82         double ventDist = 5;
83         // 방출기 위치 [m]
84         double ventHeight = 1.5;
85         // 방출기 위치 [m]
86         double ventDist = 5;
87         // 방출기 위치 [m]
88         double ventHeight = 1.5;
89         // 방출기 위치 [m]
90         double ventDist = 5;
91         // 방출기 위치 [m]
92         double ventHeight = 1.5;
93         // 방출기 위치 [m]
94         double ventDist = 5;
95         // 방출기 위치 [m]
96         double ventHeight = 1.5;
97         // 방출기 위치 [m]
98         double ventDist = 5;
99         // 방출기 위치 [m]
100        double ventHeight = 1.5;
101    }
102 }

```

Fig. 4. Set Variable Values for Simulation.

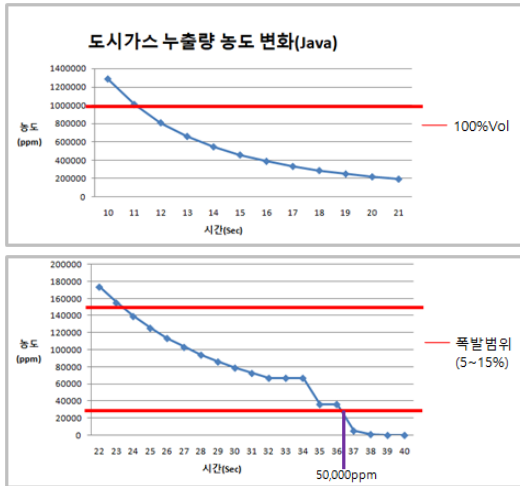


Fig 5. Graph of concentration change over time.

## V. 시뮬레이션 분석 결과

Fig 5. 은 자바 프로그램을 실행하여 데이터 값을 시간에 따른 농도(PPM)을 그래프로 나타내었다.

초기 0~10초에서는 농도(Vol)가 급격하게 낮아지지만 이미 100%(1,000,000ppm)을 초과하기에 10초 이후의 그래프를 나타내었다. 메탄의 가연성 폭발범위는 5~15% 이므로 하한계인 50,000ppm이 하가 되는 36초 이후에 위험으로부터 안전한 대기상태라고 볼 수 있다.

## VI. 결론

현재 가정 내 도시가스 누출이 되었을 경우에 농도 변화에 따른 대기위험도를 분석하여 알고리즘을 구축하고 스마트폰에 적용하고자 한다.

UI(Uesr Interface) 구현이 완료 된다면 외부 어디에 있어도 실내 가스농도를 실시간으로 확인이 가능하며 대기상태 따른 위험도를 표시하고 예측이 가능하다. 다만, 실내에서의 구조물(물건, 가구) 등을 고려될 필요가 있고 실내에서 쓰이는 물질이 다른 가스(LP, 부탄가스)를 적용하는 것도 바람직 할 것이다.

사실 JAVA를 채택한 이유 중 하나는 현재 가장 많이 쓰이는 안드로이드 스마트폰에 먼저 적용하기 위함이다. 개발이후 에는 다른 개발환경인 OS(iOS) 을 고려할 필요가 있다.

## 기호설명

- (C) : Time average concentration (mass/volume)
- $G^*$  : Total mass of material released (mass)
- $\delta x, \delta y, \delta z$  : Dispersion coefficients in the x,y and z directions(length)
- y : Cross-wind direction (length)
- z : distance above the ground
- H : release height above the ground
- $Q_m$  : Mass flow rate of gas through the hole (mass/time)
- CD : Discharge coefficient (dimensionless)
- A : Area of the hole (length<sup>2</sup>)
- PI : Pressure upstream of the hole(force/area)
- gc : Gravitataional constant (force/mass-acceleration)
- M : Molecular weight of the gas (mass/mole)
- k : Heat capacity ratio,  $C_p/C_v$  (unitless)
- Rg : Ideal gas constant (pressure-volume/mole-deg)

## 감사의 글

본 연구는 산업통산자원부의 재원으로 수행되었으며, 지원에 감사를 드립니다.

(과제번호 : 20172210100110)

## REFERENCES

- [1] Oh, W. S., Yi, W. G. Y., Oh, J. S., "Recognition of workers through deep learning (CNN) based low resolution IR image analysis", *Journal of the Korean Institute of Gas*, 22(6), 8-15, (2018)
- [2] Korea Gas Safety Corporation, 2017 Gas Statistics, (2017)
- [3] Korea Gas Safety Corporation, Gas Garage Yearbook, (2017)
- [4] Lee, S. W., and Moon, S. M., "Selective Just-in-Time Compilation for Client-side Mobile Java Script Engine," *International Conference on Compilers, Architectures and Synthesis of Embedded Systems*, ACM, 5-14, (2011)
- [5] Cha, M. S., Park, H. W., Moon, S. M., "Priority heuristics for parsing JavaScript concurrently", *Journal of Computational Sciences*, 23(8), 150-515, (2017)

- [6] *JUST JAVA Java-like Java faithful to object orientation*, Hanbit Academy, (2015)
- [7] "<http://www.eclipse.org/>", Eclipse
- [8] Kim, S. H., Lee, H. C., Park, Y. S., Lee, M. J., "Remote workspace support in the Eclipse integrated development environment", *Korean Information Science Society*, 33(2), 709-712, (2006)
- [9] Park, G. S., Yoon, I. S., *Process Safety Engineering*, (2000)
- [10] *Guidelines for Consequence Analysis of Chemical Releases*, CCPS
- [11] *Chemical Process Safety : Fundamentals with Applications*, Danel A,Crow·Joseph F, Louvar