D5.2.2: Final city search SpaceBook prototype

Morgan Fredriksson, Jürgen Königsmann, Phil Bartie, Johan Boye, Tiphaine Dalmas, Jana Goetze, Johan Mollevik, Srini Janarthanam, Oliver Lemon, Xingkun Liu, Michael Minock

Distribution: Public

SpaceBook

Spatial & Personal Adaptive Communication Environment: Behaviors & Objects & Operations & Knowledge

270019 Deliverable 5.2.2

31/03/2014

Project funded by the European Community under the Seventh Framework Programme for Research and Technological Development

The deliverable identification sheet is to be found on the reverse of this page.

Project ref. no.	270019
Project acronym	SpaceBook
Project full title	Spatial & Personal Adaptive Communication Environment: Behaviors & Objects & Operations & Knowledge
Instrument	STREP
Thematic Priority	Cognitive Systems, Interaction, and Robotics
Start date / duration	01 March 2011 / 36 Months

Security	Public
Contractual date of delivery	M34 = 30/12/2013
Actual date of delivery	31/03/2014
Deliverable number	5.2.2
Deliverable title	D5.2.2: Final city search SpaceBook prototype
Type	Report
Status & version	Final 1.0
Number of pages	9 (excluding front matter)
Contributing WP	5
WP/Task responsible	LM
Other contributors	
Author(s)	Morgan Fredriksson, Jürgen Königsmann, Phil Bartie, Johan Boye, Tiphaine Dalmas, Jana Goetze,Johan Mollevik, Srini Janarthanam, Oliver Lemon, Xingkun Liu, Michael Minock
EC Project Officer	Franco Mastroddi
Keywords	.....

The partners in SpaceBook are:

Umeå University	UMU
University of Edinburgh HCRC	UE
Heriot-Watt University	HWU
Kungliga Tekniska Högskola	KTH
Liquid Media AB	LM
University of Cambridge	UCAM
Universitat Pompeu Fabra	UPF

For copies of reports, updates on project activities and other SPACEBOOK-related information, contact:

The SPACEBOOK Project Co-ordinator:
Dr. Michael Minock
Department of Computer Science
Umeå University
Sweden 90187
mjm@cs.umu.se
Phone +46 70 597 2585 - Fax +46 90 786 6126

Copies of reports and other material can also be accessed via the project's administration homepage, `http://www.spacebook-project.eu`

Contents

Executive summary

This document describes final city search prototype for SpaceBook as well as the integrated simulated user and the interfaces that connect the two systems into one whole infrastructure.

1 Spacebook UK System Integration

The architecture of the current SpaceBook UK system is shown in figure 1. This system was integrated at HWU and is also discussed in Deliverable D1.4.

Our architecture brings together Spoken Dialogue Systems (SDS), Geographic Information Systems (GIS), and Question-Answering (QA) technologies. Its core is a spoken dialogue system (SDS) consisting of an automatic speech recogniser (ASR), a semantic parser, an Interaction Manager, a question-answering system, an utterance generator and a text-to-speech synthesizer (TTS). The GIS modules in this architecture are the City Model, the Visibility Engine and the Pedestrian tracker. Users communicate with the system using a smartphone-based client app (an Android app) that sends users' position, pace rate, and spoken utterances to the system, and delivers synthesised system utterances to the user.

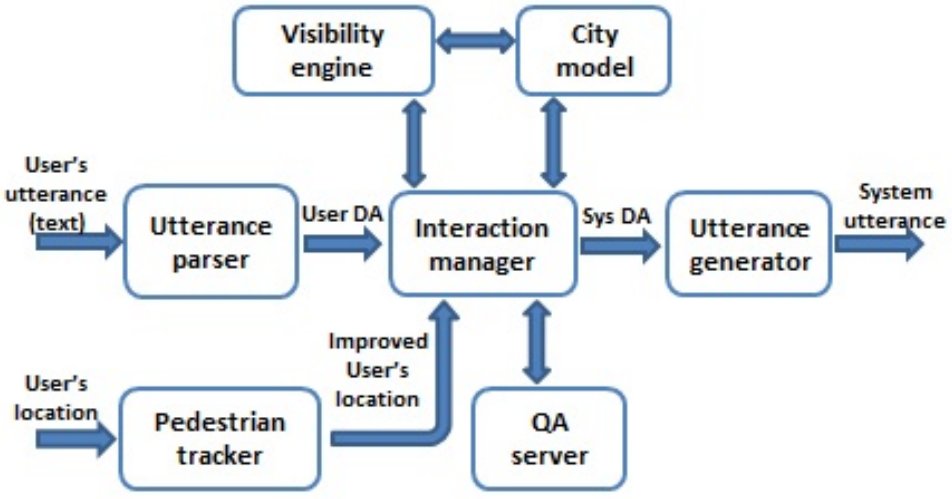

Figure 1: System Architecture

1.1 Dialogue interface

1.1.1 Speech Recognition and Parsing

The Nuance 10 speech recogniser with a domain specific language model is used for speech recognition. The language model was designed to recognise user utterances representing navigation requests (e.g. "Take me to..."), information requests (e.g. "Who was..?"', "What is..?", "Where is..?", "Tell me about..", etc), and dialogue management functions ("Can you repeat that please?"). It should be noted that the language model enables the ASR to recognise a number of variants for the same request type. For instance, the user can request directions to a destination in several ways (e.g. "Take me to X", "I want to go to X", "Directions to X" and so on). The recognised speech is then parsed into dialogue acts. Usually, a real world entity may be referred to using several name variants. For instance, "The Palace of Holyrood House" might also be colloquially be referred to as " Holyrood Palace", etc. The parser maps these name

variants in the user utterance to a unique name in the City Model. This gives the user more freedom in terms of referring to the same entity using different names.

1.1.2 Interaction Manager and dialogue policy

The Interaction Manager (IM) is the central component of this architecture, which provides the user with navigational instructions, pushes PoI information and manages QA questions. We discuss this module in detail in Deliverable D1.4.

1.1.3 Utterance generation and Speech Synthesis

The dialogue actions generated by the interaction manager are then translated into text utterances by the the utterance generation module. The generated text utterances are synthesised into speech using the CereProc Text-to-Speech server.

1.2 Pedestrian tracker

Urban environments can be challenging with limited sky views, and hence limited line of sight to satellites, in deep urban corridors. There is therefore significant uncertainty about the user's true location reported by GNSS sensors on smartphones [1]. This module improves on the reported user position by combining smartphone sensor data (e.g. accelerometer) with map matching techniques, to determine the most likely location of the pedestrian. The output includes a robust street centreline location, and a candidate space showing the probability of the user's more exact position (e.g. pavement location). This module ensures that any GNSS reported location placing the user at a rooftop location, would be corrected to the most likely ground level location, taking into consideration user trajectory history and map matching techniques (see [2]).

1.3 City Model

The City Model (CM) is a spatial database containing information about thousands of entities in the city of Edinburgh. These data have been collected from a variety of existing resources such as Ordnance Survey, OpenStreetMap, Google places and the Gazetteer for Scotland. It includes the location, use class, name, street address, and where relevant other properties such as build date and tourist ratings. The model also includes a pedestrian network (streets, pavements, tracks, steps, open spaces) which is used by an embedded route planner to calculate minimal cost routes, such as the shortest path. As a part of the route plan, the CM provides a *popularity index* for all streets in the plan. This is done by using a linear model of the number of shops (i.e. business establishments) in the street and the number of Flickr location tags on the street. The CM also provides visual descriptors of entities such as buildings and statues. These descriptors are phrases that were generated in four keys: `Is`, `Has`, `NextTo`, and `Opposite`. '`Is`' and '`Has`' keys describe the entity intrinsically. '`Is`' phrases, such as "a bronze statue of a man sitting down", describe the type of the entity with color, material and other information. Similarly, '`Has`' key phrases describe the visually salient parts of the entity such as pillars, domes, etc (e.g. "Greek pillars out the front"). `Opposite` and `NextTo` keys contain identifiers of other buildings that can used to describe the relative position of the entity in focus.

1.4 Visibility Engine

The Visibility Engine (VE) identifies the entities that are in the user's *vista space* [3]. To do this it accesses a *digital surface model*, sourced from LiDAR, which is a 2.5D representation of the city including buildings, vegetation, and land surface elevation. The VE uses this dataset to offer a number of services, such as determining the line of sight from the observer to nominated points (e.g. which junctions are visible), and determining which entities within the CM are visible and highly salient for use as landmarks in navigation instructions, determining visible entities for PoI push. A range of visual metrics are available to describe the visibility of entities, such as the field of view occupied, vertical extent visible, the facade area in view. These metrics as mentioned above are used by the Interaction Manager and the utterance generator to generate effective navigation instructions. For use as landmarks in navigation instructions, entities such as chain stores (e.g. KFC, McDonalds, Tesco, etc) that are easily identifiable by users are considered as well as buildings, statues, etc. On the other hand, for PoI push tasks, we can determine visually interesting buildings by using crowd sourced data. FlickR[1] gives us an indication of those features which are considered visually interesting, as people rarely photograph mundane items. They do however photograph the more interesting and extra-ordinary objects at either end of the beauty spectrum. In addition we use FourSquare[2] data to determine which locations are the more popular in the city and frequently visited (e.g. the Elephant House cafe).

1.5 Question-Answering server

The QA server currently answers a range of *definition* and *biographical* questions. E.g., "Tell me about the Scottish Parliament?", "Who was David Hume?", etc. QA is also capable of recognizing out of scope requests, that is, either navigation-related questions or exploration queries that cannot be handled yet ("When is the cannon gun fired from the castle?"). Question classification is entirely machine learning based using the SMO algorithm trained over 2013 annotated utterances [4].

QA then identifies the entity focused on in the question using machine-learning techniques [5]. Recognized foci include co-referring expressions, both anaphoric ("Tell me more about **him**") and deictic ("What is **this museum**?", "Who was **he**?" in front of a statue). An anaphora is resolved from the dialogue history and/or the IM context, while a deictic expression generates a query against the City Model using the estimated user view-shed and the type of point of interest identified from the user query (for instance **museum**, **statue**).

QA then proceeds to a textual search on texts from the Gazetteer of Scotland and Wikipedia. Candidate answers are reranked using a trained confidence score with the top candidate used as the final answer. These are usually long, descriptive answers and are provided as a flow of sentence chunks, one at a time. If available, QA will select the content from Wikipedia in Simple English, as it is more suitable for audio output. Otherwise, sentences will be simplified in a shallow manner (for instance, content within parentheses is removed).

The Interaction Manager also queries the QA model for pushing information based on visible and proximal points of interest. For instance, QA extracts the following sentence chunks for a query on the Royal College of Surgeons.

[1] www.flickr.com

[2] foursquare.com

"The Royal College of Surgeons is an imposing building located on Edinburgh's Nicholson Street, approximately a half-mile south of Princes Street. Designed by William Henry Playfair the foremost Scottish architect of the time, the College was opened in 1832."

1.6 Mobile client

The mobile client application, installed on an Android smartphone (Samsung Galaxy S3), connects the user to the dialogue system using a 3G data connection. The client senses the user's location using positioning technology using GNSS satellites (GPS and GLONASS) which is sent to the Interaction Manager at the rate of one update every two seconds. It also sends pace rate of the user from the accelerometer sensor. In parallel, the client also places a phone call using which the user communicates with the dialogue system.

2 Simulated User

The final Spacebook simulated pedestrians previously described in D5.3.2 and D.2.3.2, acts as a user taking instructions from a system giving natural-language route instructions. It can either be connected to such a system, or be controlled by a human operator entering natural-language instructions from the keyboard. The simulated pedestrians are capable of simulating movement in any city covered by OpenStreetMap, to understand natural language route instructions, to ask for information, and to pursue both long-term and short-term goals. The simulated walk-paths are calculated are made with a combination of the A*search algorithm and an influence map representing past behaviours of real users.

2.1 Cognitive model

In its cognitive model, the simulated user maintains a list of long-term goals. As it moves through the city, it will continuously check if its current position is close to one of those places; this mechanism mimics the desire to visit certain specific sites in the city. Apart from long-term goals it also maintains a stack of short-term goals. Finally, it maintains a representation of the set of currently visible objects. This is repeatedly retrieved from the spatial module

2.2 Spatial model

The Spatial Module creates a model of the surrounding environment by parsing an OpenStreetMap XML file. This file consist of lists of nodes defining a single lat-lon point, ways encoding roads, areas, buildings, etc., and relations between ways. From the OpenStreetMap XML file, the Spatial Module builds an object-oriented representation which forms the basis for all ensuing calculations.

2.3 Semantic parser

The semantic parser used by the simulated pedestrian is an expanded version of the parser previously described in D.4.1.1 it assigns a meaning representation to a recognized utterance. The parser is written in Prolog and works in several phases. The first step is a Definite Clause Grammar (DCG) that parses a

user utterance and assigns a preliminary semantic representation to it (rules.pl). In the second step, this representation is rewritten using some heuristic rules and a dialog act is assigned (rewrite.pl). The third step is the contextual interpretation step, where the utterance is interpreted in its geographical context. In particular, this means that references to geographical objects and locations are resolved, and relative directions like ”left” and ”right” are mapped to concrete bearings.

2.4 GUIs

The simulated user can be monitored from a graphical user interface, the current position is indicated and previous positions are traced as path together with the short term navigational goals passed by the simulated user. Line of sight queries are shown as red lines while they are calculated. A human operator can use the GUI to control the movement of the simulated user and create navigational goals for the user manually.

There is also a GUI for controlling the Simulated pedestrian from written natural language route instructions by a human operator. It shows a paraphrase of the utterances after contextual interpretation and the utterances of the simulated user together with the current coordinates and the stacked-up short-term goals are displayed.

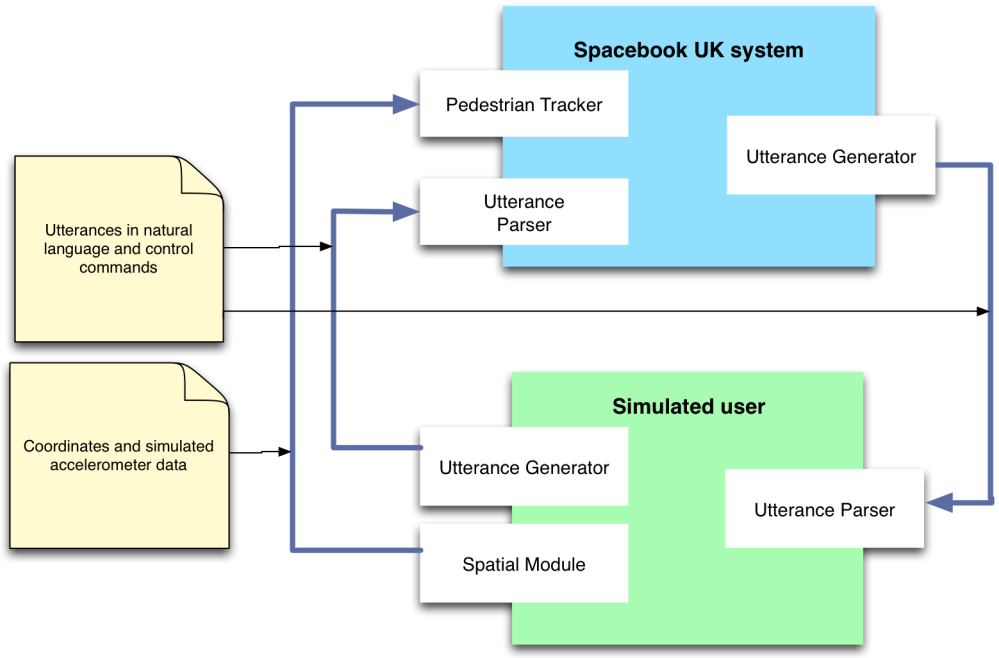

Figure 2: Communication between the Spacebook UK system and the simulated user.

2.5 Interaction between IM and simulated user

The simulated pedestrian (SP) and the Interaction Manager (IM) can interact by means of a socket interface, through which the two modules exchange JSON messages encoding utterances and other information

(like “connect”, “hangup”, etc.). When SP is switched on, it connects to the IM, and also to the Pedestrian Tracker to which coordinates and simulated accelerometer data is sent.

The messages from the IM does not only contain the actual words, but also some additional information, e.g. the *utterance type*. The SP tries to interpret all navigation instructions (utterance type is “navInstruction”). To most other utterances, the SP simply answers “okay” without further analysis, e.g. to point-of-interest information like “In front of you slightly to your left, about 70 metres away, you should be able to see School of Informatics. It is a modern glass university building.”.

After each message from the IM, the SP sends back a message indicating a “TTS finish event”, i.e. that the speech synthesizer has finished talking. The talking speed is currently assumed to be 2 words per second (e.g. for an 8-word utterance from the IM, the SP waits for 4 seconds before sending its “TTS finish event” message).

Consider the following the example interaction below. The SP’s initial position is indicated by the blue circl In this scenario, the SP was configured not to take any independent decisions on where to walk, but rather to stand still unless instructed otherwise by the IM.

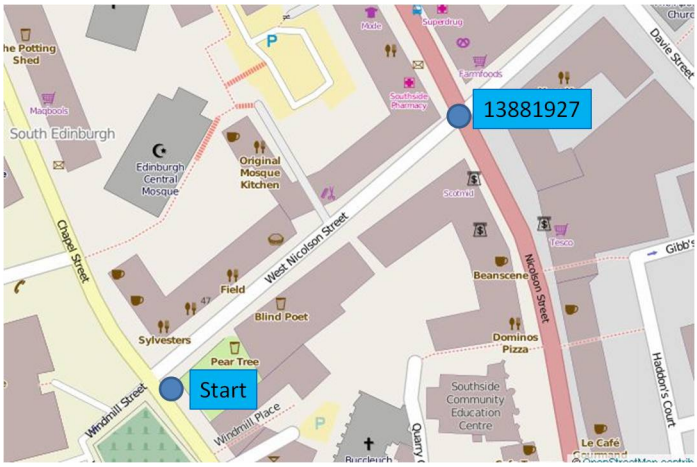

Figure 3: A routing scenario.

- SP sends a “SESSIONSTART” message.
- IM: “HELLO. MY NAME IS SPACEBOOK. I AM WAITING FOR YOUR LOCATION INFORMATION. PLEASE WAIT.”
- SP: “OKAY.”
- IM: “I AM READY. HOW CAN I HELP YOU?”
- SP: “DIRECTIONS TO SOUTHSIDER” [The SP triggers on the dialogue act of the preceding utterance being “declareReady” to state its goal.]
- IM: “OK. I AM NOW LOOKING FOR DIRECTIONS TO SOUTHSIDER.”
- SP: “OKAY.”

- IM: "YOUR DESTINATION, SOUTHSIDER, IS ABOUT 140 METRES AWAY ON WEST RICHMOND STREET."

After parsing and reference resolution, the expression representing this utterance becomes

$[\texttt{nodeOn}(\texttt{A},\texttt{B}),\texttt{isNamed}(\texttt{B},\texttt{westrichmondstreet}),\texttt{dialogAct}(\texttt{inform},\texttt{C}),\texttt{destination}(570329678),$
$\texttt{isA}(570329678,\texttt{pub}),\texttt{isNamed}(570329678,\texttt{southsider}),\texttt{C}:\texttt{distance}(\texttt{user},570329678,140)]$

The utterance is understood as an "inform" dialogue act, with its focus being `distance(user,570329678,140)`, where 570329678 is the identifier for the Southside pub. The SP stores the distance to use for future comparisons, in case the IM gives another distance figure later on.

- SP: "OKAY."
- IM: "DO YOU KNOW HOW TO GET TO WEST NICOLSON STREET?"

The expression representing this utterance is

$[\texttt{nodeOn}(\texttt{A},\texttt{B}),\texttt{isNamed}(\texttt{B},\texttt{westnicolsonstreet}),$
$\texttt{dialogAct}(\texttt{propositionalQuestion},\texttt{C}),\texttt{C}:\texttt{turn}(\texttt{user},\texttt{D},\texttt{A},\texttt{E},\texttt{F})]$

The SP's understanding of the utterance could be paraphrased as "Is it possible for you to turn and go to `A`, where `A` is a node on West Nicolson Street?". The reference resolution component then finds a node `A` which fulfills the contraints, in this case 1989980580.

- SP: "YES"
- IM: "WALK ALONG NILE VALLEY CAFE AND RED BOX NOODLE BAR ON YOUR LEFT, AND PEAR TREE ON YOUR RIGHT."

The SP cannot make sense of this utterance, and says nothing.

- IM (repeating): "WALK ALONG NILE VALLEY CAFE AND RED BOX NOODLE BAR ON YOUR LEFT, AND PEAR TREE ON YOUR RIGHT."
- IM: "TURN RIGHT ONTO WEST NICOLSON STREET"

Represented as:

$[\texttt{nodeOn}(\texttt{A},\texttt{B}),\texttt{isNamed}(\texttt{B},\texttt{westnicolsonstreet}),$
$\texttt{C}:\texttt{turn}(\texttt{user},\texttt{right},\texttt{D},\texttt{A},\texttt{E}),\texttt{dialogAct}(\texttt{instruct},\texttt{C})]$

After reference resolution, the SP instantiates `A` to 13881927, which is the node in the intersection of West Nicolson Street, Nicolson Street, and West Richmond Street (see 3).

- The SP starts moving towards node 13881927.
- IM: "IN FRONT OF YOU SLIGHTLY TO YOUR LEFT, ABOUT 90 METRES AWAY, YOU SHOULD BE ABLE TO SEE SCHOOL OF INFORMATICS. IT IS A MODERN GLASS UNIVERSITY BUILDING."
- SP: "OKAY"

References

[1] P. A. Zandbergen and S. J. Barbeau. Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. *Journal of Navigation*, 64(3):381399.

[2] P. Bartie and W. Mackaness. D3.4 pedestrian position tracker. *Technical report, The SPACEBOOK Project (FP7/2011-2014 grant agreement no. 27001)*, 2012.

[3] D. Montello. Scale and multiple psychologies of space. *Spatial information theory: A theoretical basis for GIS*, pages 312–321.

[4] C. Bhattacharyya S.S. Keerthi, S. K. Shevade and K. R. K. Murthy. "Improvements to Platts SMO algorithm for SVM classifier design.". *Neural Computation*, (3):637649.

[5] T. Dalmas A. Mikhailsian and R. Pinchuk. Learning foci for question answering over topic maps. *Proceedings of ACL*, 2009.