



**ARISTOTLE UNIVERSITY OF THESSALONIKI**  
School of Electrical & Computer Engineering

# **“Integrating Social and Hardware Sensors: A Semantic Web framework for Social Sensing environments”**

A thesis submitted in fulfilment for the degree of Doctor of Philosophy

**IOANNIS PAGKALOS**

**2017**





**ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣ/ΝΙΚΗΣ**  
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

**“Πλαίσιο Αρχιτεκτονικής  
Σημασιολογικού Ιστού για την  
ενσωμάτωση αισθητήρων Κοινωνικής  
Δικτύωσης και αισθητήρων Υλικού”**

Διδακτορική Διατριβή του

**ΙΩΑΝΝΗ ΠΑΓΚΑΛΟΥ**

**2017**





## "ABSTRACT"@en

---

The amount of content that Social Media users produce nowadays is staggering. In Social Media platforms, such as Facebook and Twitter, *humans* act implicitly as *sensors*, making *observations* about real-world events or themselves that are extremely varied; from their opinion on what the weather is like, to what is happening out of their window, to how many miles they ran today in their daily workout routine. In a way, these users can be treated as **Social Sensors**: a source of self-reported data that resembles sensor observations about a particular event or property and is accompanied by a wealth of social metadata, both very useful to a variety of scientific disciplines (e.g. e-Health, nutrition & dietetics, social sciences), especially when *integrated* with data from traditional hardware sensors.

Nevertheless, data from social sensors is mostly unstructured, untrusted and usually provided without the provenance chains that are required for assessing data quality and, eventually, decision making. Semantic Web technologies can play a beneficial role in this emerging area. More specifically, ontologies, which are typically represented on the Semantic Web via OWL, the Web Ontology Language, clarify the domain's structure of knowledge and enable knowledge-sharing, a critical component in Social Sensing.

This thesis presents the **SOSENS** (SOcial **SENS**or) framework, an ontology-based solution which provides a “semantic scaffold” for Social Sensing spaces. The basic ontology (SOSENS) combines several well-established ontologies to describe humans as Sensors on Social Media in a systematic manner, bringing them to the same semantic level as a Hardware Sensor and allowing the integration of the two when needed. Extensions, in the forms of ontologies such as SOSENS-Trust, provide the groundwork for describing other aspects of Social Sensing spaces, such as Social Sensor data quality and trust.

The framework was evaluated in the NutriHeAl e-Health project, where self-reported physical activity from Facebook Social Sensors is combined with Fitbit Digital Pedometers in order to track physical activity as well as calculate data quality and user trust. Using the framework, the semantically-rich data of this complex sensing space was efficiently collected, integrated, analysed and stored along with clear provenance & attribution metadata of all the actors & processes involved. SOSENS improves information quality, fosters reusability and creates interoperable social sensing spaces for the Web 3.0 era.

### Εισαγωγή στο αντικείμενο

Η ποσότητα της πληροφορίας που παράγεται σήμερα από τους χρήστες των μέσων Κοινωνικών Δικτύωσης (Social Media) έχει ξεπεράσει κάθε προσδοκία. Οι μηνιαίοι χρήστες του Facebook έχουν φτάσει το 1.7+ δισεκατομμύριο ενώ οι χρήστες του Twitter τα 315 εκατομμύρια. Σε τέτοια περιβάλλοντα, οι χρήστες δρουν συχνά ως *αισθητήρες*, πραγματοποιώντας ποικίλες μετρήσεις και καταγράφοντας έναν εντυπωσιακό όγκο από δεδομένα για το περιβάλλον τους ή ακόμα και για τους ίδιους που είναι εξαιρετικά χρήσιμα σε πληθώρα επιστημών (π.χ. ηλεκτρονική υγεία, διατροφή & άσκηση, κοινωνικές επιστήμες): από την άποψή τους για τον καιρό ή για το τι συμβαίνει στο άμεσο περιβάλλον τους, μέχρι το πόσα χιλιόμετρα έτρεξαν στην καθημερινή τους εξάσκηση.

Στην επιστημονική βιβλιογραφία, η έννοια του Ανθρώπινου Αισθητήρα (Human Sensor) έχει μελετηθεί σε πληθώρα επιστημονικών περιοχών όπως τα συστήματα Εθελοντικής Γεωγραφικής Πληροφορίας (Voluntary Geographic Information – VGI), την Ηλεκτρονική Άμυνα & Ασφάλεια (defense & security, όπου η ανθρώπινη πληροφορία χαρακτηρίζεται ως HUMINT – Human Intelligence), το χώρο του Pervasive & ubiquitous Computing καθώς και το χώρο της (ηλεκτρονικής) Υγείας (e-)Health, όπου η πληροφορία που προέρχεται από το χρήστη είναι, πολλές φορές, το πιο χρήσιμο εργαλείο στα χέρια ενός επιστήμονα. Με την «έκρηξη» του Web 2.0, η έννοια του Human Sensor απέκτησε νέες πτυχές, με ερευνητές όπως οι Sheth & Goodchild να οριοθετούν νέες περιοχές όπως το Citizen Science. Τα τελευταία χρόνια, η πρωτοφανής επιρροή των Social Media στο Web ώθησε την επιστημονική κοινότητα να μελετήσει το φαινόμενο του «Social Sensing»: ένας όρος που δημιουργήθηκε για να περιγράψει τους χρήστες των Social Media ως **«Αισθητήρες Κοινωνικής Δικτύωσης» (Social Sensors)**.

Η πρωταρχική μελέτη του φαινομένου έγινε από τον Sakaki, όταν το 2010 δημοσίευσε την εργασία «Earthquake shakes Twitter users: real-time event detection by social sensors», στην οποία παρατήρησε ότι οι χρήστες του Twitter ήταν ικανότατοι αισθητήρες σεισμών και μπόρεσε μέχρι και να προσδιορίσει το επίκεντρο ενός σεισμού από την γεωγραφική τοποθεσία (geolocation) των tweets. Έκτοτε, ερευνητές έχουν διερευνήσει τη χρήση των Social Sensors σε ποικίλες περιστάσεις: από παρόμοια συμβάντα φυσικών καταστροφών (natural disasters), στην real-time αίσθηση συμβάντων σε αγώνες (sports event detection), έως και την αναγνώριση “trends” στο χρηματιστήριο. Στο χώρο της ηλεκτρονικής Υγείας, μια επιστήμη με ξεκάθαρη κοινωνική διάσταση όσον αφορά τη διάχυση πληροφορίας,

ερευνητές παρακολούθησαν την εξέλιξη επιδημιολογικών φαινομένων στο twitter όπως ο ιός Influenza H1N1 στην Αγγλία, με βαθμό επιτυχίας που άγγιξε την 95% συσχέτιση της πληροφορίας από tweets με τα σχετικά αρχεία της επίσημης υπηρεσίας (UK Health Protection Agency).

Παρ' όλα αυτά, τα δεδομένα που παράγει ένας Social Sensor είναι, κατά κύριο λόγο, μη δομημένα, κυμαινόμενης αξιοπιστίας και χωρίς τις απαραίτητες αλυσίδες προέλευσης (provenance chains) που απαιτούνται για την εξακρίβωση της ποιότητας και την αξιοποίησή τους. Η σύγκριση της πληροφορίας με δεδομένα από αισθητήρες υλικού (hardware sensors) είναι συχνά απαραίτητη αλλά η ετερογένεια των δύο πηγών καθιστά τις περισσότερες φορές την εφαρμογή τέτοιων τεχνικών μη εφικτή. Ως επί το πλείστον, αποτελέσματα από περιβάλλοντα Social Sensing είναι δομημένα σε μοντέλα άμεσα συνδεδεμένα με την εφαρμογή (application-specific) και «εγκλωβισμένα» σε «σιλό» πληροφορίας (information silos). Πρόσφατες ερευνητικές μελέτες (research roadmaps) τόνισαν την ανάγκη για τη δημιουργία μοντέλων γύρω από το χώρο του Social Sensing που θα είναι επαναχρησιμοποιήσιμα και αγνωστικά της περιοχής (domain-independent).

Βασικός **στόχος** της ερευνητικής αυτής εργασίας είναι να αποδείξει ότι οι **τεχνολογίες Σημασιολογικού Ιστού (Semantic Web)** μπορούν να χρησιμοποιηθούν για τη λύση αυτών των προβλημάτων, μετατρέποντας την πληροφορία των Social Sensors σε μορφή που είναι καλύτερα κατανοητή και επεξεργάσιμη τόσο από τον άνθρωπο όσο και από μία μηχανή και, ταυτόχρονα, επιτρέποντας τον συνδυασμό και την ενοποίησή της με πληροφορία από πραγματικούς αισθητήρες.

## **Διάρθρωση της Διατριβής**

Τα βασικά ερευνητικά θέματα που τίθενται στην διατριβή αυτή είναι η εξερεύνηση του ρόλου των τεχνολογιών Σημασιολογικού Ιστού στο νέο, συνεχώς εξελισσόμενο χώρο του Social Sensing, ο σχεδιασμός μίας κατάλληλης αρχιτεκτονικής για τη διαχείριση των δεδομένων, και η εφαρμογή και αξιολόγησή της σε ένα πραγματικό πρόβλημα Social Sensing. Για το σκοπό αυτό, η διατριβή αυτή επικεντρώθηκε στις παρακάτω βασικές ενότητες:

**Κεφάλαιο 1 (Εισαγωγή):** Στο κεφάλαιο αυτό γίνεται μια σύντομη περιγραφή του προβλήματος του Social Sensing. Στην συνέχεια παρουσιάζεται η συνεισφορά της διδακτορικής διατριβής και τίθενται οι ερευνητικοί στόχοι της. Τέλος, περιγράφεται η δομή της διατριβής σε τμήματα και κεφάλαια.

**Κεφάλαιο 2 (Επιστημονική Ανασκόπηση):** Στο κεφάλαιο αυτό παρουσιάζονται οι βασικές έννοιες καθώς και οι τελευταίες ερευνητικές εξελίξεις στο χώρο των

Human Sensing, Social Media και Social Sensing. Δίνονται ακόμα παραδείγματα από πειράματα Social Sensing, από τα οποία γίνεται σαφής η ανομοιογένεια του χώρου καθώς και η ανάγκη για ένα καθολικό μοντέλο που θα είναι αγνωστικό της εφαρμογής. Στη συνέχεια γίνεται μία εκτενής εισαγωγή στις τεχνολογίες Semantic Web (RDF, ontologies, querying) και πως αυτές μπορούν να συμβάλλουν στη μοντελοποίηση και ομοιογένεια ενός περιβάλλοντος Social Sensing.

**Κεφάλαιο 3 (Οντολογίες):** Κατά τη διάρκεια της διατριβής, σχεδιάστηκε και υλοποιήθηκε μία υψηλού επιπέδου (high-level) οντολογία με τίτλο SOSENS (*the **S**ocial **SENS**or ontology*). Η οντολογία αυτή χρησιμοποιεί ως βάση state-of-the-art οντολογίες όπως η FOAF (Friend-of-a-Friend), SIOC (Semantically-Interlinked Online Communities) και SSN (Semantic Sensor Network), ακολουθώντας την ίδια αφηρημένη προσέγγιση, με απώτερο σκοπό την δημιουργία ενός «σημασιολογικού σκελετού» για την περιγραφή ενός Social Sensing περιβάλλοντος. Χρησιμοποιώντας την SOSENS, οι Social Sensors περιγράφονται σημασιολογικά με τον ίδιο τρόπο όπως και οι «κλασσικοί» αισθητήρες υλικού (hardware sensors) και οι μετρήσεις τους συνδέονται με το πηγαίο Social Media αντικείμενο, δημιουργώντας έτσι τις απαραίτητες αλυσίδες προέλευσης που μπορούν να χρησιμοποιηθούν με πολλαπλούς τρόπους σε ένα περιβάλλον Σημασιολογικού Ιστού.

Με τη χρήση της οντολογίας αυτής δίνεται για πρώτη φορά η δυνατότητα στους χρήστες ενός τέτοιου περιβάλλοντος (είτε αυτοί είναι άνθρωποι είτε «μηχανές» που έχουν διεπαφή με τον Παγκόσμιο Ιστό) να πάρουν ταυτόχρονα, με απλά ερωτήματα τύπου SPARQL, πληροφορίες τόσο από Social όσο και από Hardware sensors, καθώς και να θέσουν σύνθετα ερωτήματα με βάση το υπάρχον κοινωνικό δίκτυο.

Όπως είναι αναμενόμενο, τα δεδομένα στα μέσα Κοινωνικής Δικτύωσης χαρακτηρίζονται από υψηλό επίπεδο «θορύβου». Για την αξιοποίησή τους, συνεπώς, από ανθρώπινους χρήστες και υπολογιστές αντίστοιχα, είναι απαραίτητο να υπάρχει κάποιος δείκτης αξιοπιστίας που να βασίζεται σε μια αντίστοιχα αξιόπιστη διαδικασία υπολογισμού. Για το σκοπό αυτό, η διατριβή αυτή ανέπτυξε και εφάρμοσε επεκτάσεις της βασικής οντολογίας SOSENS, όπως την SOSENS-Trust, που παρέχει το απαραίτητο σημασιολογικό μοντέλο για την περιγραφή παραμέτρων αξιοπιστίας.

**Κεφάλαιο 4 (Αρχιτεκτονική):** Στο πλαίσιο της διατριβής αναπτύχθηκε μία πλήρης αρχιτεκτονική, τόσο σε επίπεδο σχεδιασμού (Architecture Building Blocks) όσο και υλοποίησης (Software Building Blocks) για την αξιοποίηση των οντολογιών SOSENS, με τον ομώνυμο τίτλο (*the **SOSENS** Framework*). Η αρχιτεκτονική αυτή

συνοδεύεται από μία ισχυρή και ευέλικτη διαδικτυακή διεπαφή (Web API - Application Programming Interface) που διευκολύνει την υλοποίηση εφαρμογών σε περιβάλλοντα Social Sensing. Το Web API είναι σχεδιασμένο τόσο για χρήστες ειδικούς στις τεχνολογίες Σηματολογικού Ιστού όσο και για μη-ειδικούς, καθιστώντας τη χρήση του SOSENS και τη δημιουργία σηματολογικά-πλούσιων δεδομένων σημαντικά πιο εύκολη. Ένα demo του API καθώς και πληροφορίες σχετικά με το SOSENS framework υπάρχει μόνιμα διαθέσιμο στη διεύθυνση <http://phd.pagkalos.com/sosens>.

**Κεφάλαιο 5 (Πιλοτική υλοποίηση):** Η διατριβή παρουσιάζει επίσης μία πλήρη υλοποίηση (full stack) της προτεινόμενης αρχιτεκτονικής SOSENS και αναλύει τα δεδομένα που συγκεντρώθηκαν κατά τη διάρκεια της εφαρμογής της σε ένα πρωτότυπο, πραγματικό περιβάλλον Social Sensing της ανθρώπινης δραστηριότητας (human activity) στο Facebook. Ένα δείγμα 49 χρηστών (ηλικίας  $24 \pm 7$ ) συμμετείχε στην πιλοτική υλοποίηση για περισσότερες από 30 ημέρες και παρείχε δεδομένα μέσω μίας ειδικής εφαρμογής στο Facebook που αναπτύχθηκε στα πλαίσια της διατριβής. Πιο συγκεκριμένα:

- Συλλέχθηκαν αυτό-αναφερόμενες (self-reported) μετρήσεις στο Facebook σχετικά με τη φυσική δραστηριότητα των χρηστών και εκφράστηκαν ως *Social Sensor Observations*
- Συλλέχθηκαν δεδομένα για το ίδιο διάστημα από πραγματικούς αισθητήρες (ψηφιακά βηματόμετρα Fitbit Zip)

Ταυτόχρονα, σύμφωνα με την μεθοδολογία προσδιορισμού αξιοπιστίας του προγράμματος υπολογίσθηκαν:

- Ο βαθμός αξιοπιστίας κάθε μέτρησης από το συνδυασμό των δύο πηγών μέσω ασαφών συναρτήσεων συμμετοχής (Fuzzy Membership Functions) και
- Ο βαθμός εμπιστοσύνης κάθε χρήστη ως Activity Social Sensor μέσω εξισώσεων πολωνυμικής φήμης Beta (Multinomial Beta Reputation)

Όλο το περιβάλλον Social Sensing και υπολογισμού αξιοπιστίας περιγράφηκε και υλοποιήθηκε μέσω του SOSENS (και των επεκτάσεών του), δημιουργώντας έτσι έναν πλούτο επαναχρησιμοποιήσιμου υλικού και διαδικασιών. Το κεφάλαιο αυτό δρα και σαν «hands-on demo» για τη χρήση του framework, καθώς συζητούνται σε βάθος οι ιδιαιτερότητες του συγκεκριμένου Social Sensing περιβάλλοντος και πως αυτές περιγράφονται μέσω του SOSENS.

**Κεφάλαιο 6 (Συμπεράσματα):** Στο τελικό κεφάλαιο της διατριβής, γίνεται μία σύντομη ανασκόπηση της και παρουσιάζονται τα κυριότερα συμπεράσματα και η συμβολή της στο χώρο του Social Sensing. Τέλος, αναφέρονται τα σημεία της

έρευνας που θα μπορούσαν να αποτελέσουν αντικείμενο μελλοντικής ερευνητικής δραστηριότητας.

Η διατριβή συνοδεύεται από δύο παραρτήματα:

Στο **Παράρτημα Α** παρουσιάζονται και συζητούνται τα αποτελέσματα της πιλοτικής υλοποίησης και τα οφέλη στο πεδίο εφαρμογής της Ηλεκτρονικής Υγείας.

Στο **Παράρτημα Β** παρουσιάζεται δείγμα των δυνατοτήτων του SOSENS Web API με παραδείγματα για τη δημιουργία Social & Hardware Sensor observations.

## Συμβολή της Διατριβής

Τα βασικά σημεία της επιστημονικής συμβολής της διατριβής συνοψίζονται στα παρακάτω:

- Στην παρούσα διατριβή, τα θέματα σχεδιασμού και υλοποίησης αντιμετωπίζονται εξ' ολοκλήρου από την οπτική των τεχνολογιών *Σημασιολογικού Ιστού* που αποτελεί και την πρώτη αντίστοιχη *συστηματική* ερευνητική προσπάθεια στον τομέα του Social Sensing.
- Χρησιμοποιώντας την ανεπτυγμένη οντολογία *SOSENS*, ερευνητές ανά τον κόσμο μπορούν να εκφράσουν τα δεδομένα ενός Social Sensing χώρου με έναν ομοιόμορφο τρόπο που επιτρέπει:
  - Την κατανόησή τους τόσο από ανθρώπους όσο και από μηχανές (machine-understandable semantics)
  - Την ενοποίησή τους και σύγκρισή τους με δεδομένα από άλλους χώρους Sensing αλλά και με «κλασσικούς» αισθητήρες υλικού
  - Την περιγραφή των δεδομένων από πολύτιμα μετα-δεδομένα (πηγαίο αντικείμενο, διαδικασία εξόρυξης κτλ) που ανεβάζουν το δείκτη ποιότητας της πληροφορίας, επιτρέποντας στους χρήστες να επιλέγουν καλύτερα τα δεδομένα που τους ενδιαφέρουν
- Η οντολογία *SOSENS* είναι, από τη φύση της, εύκολα επεκτάσιμη, κάτι το οποίο παρουσιάστηκε στα πλαίσια της διατριβής με την οντολογία *SOSENS-Trust* που προσθέτει τη δυνατότητα για την καταγραφή τιμών και διαδικασιών αξιολόγησης του βαθμού εμπιστοσύνης της πληροφορίας.
- Στα πλαίσια της διατριβής σχεδιάστηκε ένα πλήρες πλαίσιο αρχιτεκτονικής (architecture framework) με βάση την «οικογένεια» οντολογιών *SOSENS*:

- *To SOSENS framework*, ακολουθώντας τα πρότυπα του the Open Group Architecture Forum (TOGAF) αναλύεται με τη μορφή διακριτών Architecture Building Blocks που επιτρέπουν τις διαφορετικές υλοποιήσεις του ανάλογα με τις ανάγκες ενός Social Sensing χώρου.
- Παρουσιάστηκε, ταυτόχρονα, ένα Reference Implementation, με προτεινόμενα Software Building Blocks, όπως το SOSENS Web API που δημιουργήθηκε στα πλαίσια της διατριβής για τη διευκόλυνση της δημιουργίας σημασιολογικά πλούσιου περιεχομένου από τους χρήστες
- Το μεγαλύτερο μέρος του λογισμικού που δημιουργήθηκε στα πλαίσια της διατριβής απευθύνεται τόσο σε ειδικούς, όσο και σε μη-ειδικούς και διατίθεται τμηματικά προς δημόσια χρήση, καθιστώντας το έτσι χρήσιμο σε μια σειρά από επιστημονικές περιοχές που ασχολούνται ή μπορούν να επωφεληθούν από το Social Sensing.
- Η διατριβή επιβεβαιώνει το σημαντικό ρόλο των τεχνολογιών Σημασιολογικού Ιστού στον τομέα και αποδεικνύει την καταλληλότητα του προτεινόμενου *SOSENS framework* μέσα από την υλοποίηση ενός *πilotου* σε πραγματικό περιβάλλον ο οποίος μπορεί να αποτελέσει τη βάση για αντίστοιχες εφαρμογές.
  - Η πιλοτική υλοποίηση - στο χώρο της Ηλεκτρονικής Υγείας – απέδειξε την πληθώρα οφελών του Framework: εύκολη ενοποίηση πληροφορίας από Facebook Social Sensors & Fitbit Digital Pedometers, μοντελοποίηση και καταγραφή των διαδικασιών αξιολόγησης αξιοπιστίας Fuzzy Membership & Beta Reputation, εύκολη σύνδεση της πληροφορίας με άλλες βάσεις δεδομένων κτλ.
  - Τα πλεονεκτήματα αυτά *δεν περιορίζονται* στο χώρο του e-Health καθώς προκύπτουν από τις *εγγενείς* δυνατότητες της αρχιτεκτονικής SOSENS, λόγω της χρήσης τεχνολογιών Σημασιολογικού Ιστού.
- Η διατριβή, λόγω της ξεκάθαρα σύγχρονης προσέγγισής της, οριοθετεί το χώρο του Social Sensing στο «έξυπνο» περιβάλλον Web 3.0 και συμβάλλει, ταυτόχρονα, σε νέους ερευνητικούς χώρους όπως η Επιστήμη του Διαδικτύου (Web Science) και η Διασυνδεδεμένη Έρευνα (Linked Research).

## ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

**Κανόνες (Rules Inferencing):** Στην παρούσα φάση, δεν υπάρχει κάποιο Semantic Web standard για την περιγραφή και εφαρμογή *κανόνων (Rules Inferencing)*, παρ'οτι υπάρχουν πολλές λύσεις εντός του technology stack (RIF, SWRL, SPIN).

Δεδομένης της αυξημένης σημασίας που λαμβάνει το θέμα εντός της επιστημονικής κοινότητας του Semantic Web τον τελευταίο καιρό, είναι αναμενόμενο ότι σύντομα θα προκύψει κάποιο standard. Στην περίπτωση αυτή, μία χρήσιμη επέκταση του SOSENS θα ήταν η προσθήκη κανόνων εντός του framework για τον υπολογισμό της αξιοπιστίας των χρηστών απευθείας από τη μηχανή inferencing. Για παράδειγμα, οι τεχνικές Fuzzy MF & mBRS που χρησιμοποιήθηκαν στα πλαίσια του πιλότου θα μπορούσαν να παρέχονται ως pre-built inferencing rules, κάτι το οποίο θα αυξήσει τις out-of-the-box δυνατότητες της αρχιτεκτονικής

**Ιδιωτικότητα (Privacy):** Δεδομένης της σύνθετης φύσης ενός χώρου Κοινωνικής Δικτύωσης, υπάρχει ένα εγγενές πρόβλημα ιδιωτικότητας το οποίο μπορεί να λυθεί με την εφαρμογή αλγορίθμων όπως ο k-anonymity. Εντός του framework, τέτοιες τεχνικές μπορούν να προστεθούν σε πολλαπλά σημεία, ανάλογα με τις κατέκαστες ανάγκες ιδιωτικότητας, ακόμα και σε μορφή κανόνων.

**Εφαρμογές Semantic Social Sensing:** Το SOSENS framework συμβάλλει στη δημιουργία σημασιολογικά πλούσιας πληροφορίας, η οποία είναι ακόμα πιο χρήσιμη αν υπάρχουν αντίστοιχες εφαρμογές για την αξιοποίησή της. Κάτι που θα συμβάλλει σε αυτό είναι η δημιουργία νέων, Semantic-Web aware εφαρμογών εντός του χώρου, όπως για παράδειγμα μία μηχανή οπτικοποίησης και αξιολόγησης των κοινωνικών δεσμών μεταξύ των χρηστών (Social Network map) που θα λαμβάνει υπόψιν και πληροφορία που προέρχεται από μεταδεδομένα όπως η αξιοπιστία τους.



## "ACKNOWLEDGMENTS"@en

---

I'd like to thank my supervisor, Prof. Loukas Petrou, for keeping an open mind towards every single “ground-breaking, paradigm-shifting” idea I had and keeping me level all these years. His paramount attention to detail and calm demeanour is one of the main reasons why I managed to complete this work.

I'd also like to thank Prof. Mark Weal who, apart from being an advisor to this thesis, was also an important advisor to me during the formative years of my MSc Degree, in the university of Southampton. Additionally, I'd like to thank Prof. Sokratis Katsikas and the members of the examination committee for reviewing and evaluating this work.

I truly have to thank my parents, who, although academics themselves, managed to stay (relatively) away from my thesis, even though I probably couldn't have done it without them (or would have strived for a Doctoral Degree in the first place). I must thank my sister, too, for excelling in everything I'd be too “rational” to even attempt and changing my views on so many things, despite my older brother “+4 Wisdom” bonus.

Finally, I'd like to thank all the members of the lab in the Aristotle University for enduring the endless discussions which consisted mainly of me talking about how “cool” earthquakes on Twitter were as well as all my friends and co-workers who suffered the same fate at one point or another and became idea-bouncing engines for many of the topics discussed within.

## DIGITAL Download

---

You can obtain a PDF version of this thesis by visiting the following link: <http://phd.pagkalos.com/thesis/thesis.pdf> or by scanning the QR code below:





# Table of Contents

---

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	MOTIVATIONAL SCENARIO .....	1
1.2	THE RESEARCH PROBLEM .....	3
1.3	RESEARCH QUESTIONS .....	3
1.4	RESEARCH GOALS & DOCUMENT OUTLINE .....	5
<b>2</b>	<b>THE RESEARCH SPACE .....</b>	<b>7</b>
2.1	HUMANS AS SENSORS .....	7
2.2	SOCIAL MEDIA: A VAST REPOSITORY OF HUMAN SENSOR OBSERVATIONS....	9
2.3	SOCIAL SENSING .....	11
2.4	(RE-)DEFINING THE PROBLEM .....	15
2.5	THE SEMANTIC WEB .....	17
<b>3</b>	<b>MODELLING SOCIAL SENSING ENVIRONMENTS .....</b>	<b>35</b>
3.1	MODELLING HARDWARE SENSORS .....	35
3.2	MODELLING SOCIAL SENSORS .....	42
3.3	THE SOCIAL SENSOR (SOSENS) ONTOLOGY .....	48
3.4	SOSENS EXTENSIONS .....	53
<b>4</b>	<b>THE SOSENS FRAMEWORK .....</b>	<b>61</b>
4.1	FRAMEWORK OVERVIEW .....	62
4.2	SOSENS ARCHITECTURE BUILDING BLOCKS (ABBs) .....	64
4.3	SAMPLE SOSENS SOLUTION BUILDING BLOCKS (SBBS).....	69
4.4	EVALUATION .....	73
<b>5</b>	<b>PILOT IMPLEMENTATION .....</b>	<b>81</b>
5.1	THE “NUTRIHEAL EXERCISE SOCIAL SENSING” PILOT .....	81
5.2	DATA COLLECTION .....	87
5.3	USING SOSENS TO MODEL COLLECTED DATA .....	89
5.4	DATA QUALITY ASSESSMENT (DQA) .....	97
5.5	BENEFITS OF SOSENS .....	120
<b>6</b>	<b>CONCLUSIONS .....</b>	<b>129</b>
6.1	CONTRIBUTIONS .....	130
6.2	RESEARCH PUBLICATIONS .....	131
6.3	FUTURE WORK .....	132
	<b>REFERENCES .....</b>	<b>137</b>

<b>APPENDIX A .....</b>	<b>147</b>
THE NUTRIHEAL PILOT.....	147
<b>APPENDIX B .....</b>	<b>153</b>
THE SOSENS API.....	153

## List of Figures

---

<b>Figure 1:</b> An example query to a Social Sensing Space.....	2
<b>Figure 2:</b> A diagram of popular Social Media. ....	10
<b>Figure 3:</b> Earthquake estimation by user tweets .....	14
<b>Figure 4:</b> An example Social Sensor post on Facebook.....	16
<b>Figure 5:</b> Critical information extracted from a social sensor post on Facebook .....	18
<b>Figure 6:</b> The “walled gardens” of the early 2000s era of Social Networking.....	19
<b>Figure 7:</b> The Semantic Web stack .....	20
<b>Figure 8:</b> Statements in RDF .....	21
<b>Figure 9:</b> An example RDF Graph .....	22
<b>Figure 10:</b> Federated SPARQL query example .....	30
<b>Figure 11:</b> The BodyMedia SenseWear Sensor & its sensing capabilities	36
<b>Figure 12:</b> Sensor data & metadata to be modelled in a sensing scenario .....	37
<b>Figure 13:</b> ssn:Sensor and ssn:MeasurementCapability .....	39
<b>Figure 14:</b> Implementation of the Stimulus-Sensor-Observation design pattern in the SSN .....	40
<b>Figure 15:</b> An example observation in SSN .....	41
<b>Figure 16:</b> Examples of Social Sensor data & metadata to be modelled ..	43
<b>Figure 17:</b> SIOC + FOAF .....	46
<b>Figure 18:</b> The SIOC core ontology .....	47
<b>Figure 19:</b> Describing a Social Sensor observation on Facebook using the Social Sensor (SOSENS) Ontology .....	51
<b>Figure 20:</b> A categorisation of major areas of trust research [90]. Numbers in parentheses indicate the amount of research papers associated with each sub-category .....	55
<b>Figure 21:</b> Describing observation quality in SOSENS-T .....	57
<b>Figure 22:</b> Describing Sensor Trust in SOSENS-T .....	59
<b>Figure 23:</b> Overview of the SOSENS Framework.....	62
<b>Figure 24:</b> Sample open-source & commercial Triplestores .....	69
<b>Figure 25:</b> Converting a JSON input from a Hardware Sensor (top) to OWL/N3 (bottom) using the SOSENS API (simple mode) .....	72
<b>Figure 26:</b> The 15 most important data quality dimensions from consumer perspective. ....	74
<b>Figure 27:</b> Sample response times for common SOSENS procedures.....	77
<b>Figure 28:</b> The SOSENS public website .....	78

<b>Figure 29:</b> The SOSENS public website – API Demo.....	79
<b>Figure 30:</b> The NutriHeAI Facebook app – Profile Tab .....	84
<b>Figure 31:</b> The NutriHeAI Facebook app - Activity Diary tab .....	86
<b>Figure 32:</b> Example of a result Graph.....	87
<b>Figure 33:</b> Some of the manufacturer-approved locations for wearing the Fitbit Zip .....	88
<b>Figure 34:</b> Modelling a Social Observation from the ESS pilot using SOSENS .....	92
<b>Figure 35:</b> Membership functions for Low+ and Moderate+ Exercise Activities using the OQ_1 Metric.....	100
<b>Figure 36:</b> Membership functions for Low+ and Moderate+ Exercise Activities using the OQ_2 Metric.....	101
<b>Figure 37:</b> An excerpt from the FuzzyMF ontology. Screenshot from TopBraid Composer.....	103
<b>Figure 38:</b> Using SOSENS-T to model Social Sensor Observation Quality in the ESS Pilot.....	105
<b>Figure 39:</b> Beta function of event after 7 positive and 1 negative outcome. .....	108
<b>Figure 40:</b> Disparity between users' expected quality as captured by a mBRS system. ....	109
<b>Figure 41:</b> Fuzzy triangular membership functions.....	110
<b>Figure 42:</b> Modelling Social Sensor Trust in SOSENS-T using a 5-level mBRS system. Screenshot from TopBraid Composer.....	116
<b>Figure 43:</b> Using SOSENS-T to model Social Sensor Trust in the ESS Pilot .....	118
<b>Figure 44:</b> Social Network Analysis of the NutriHeAI Pilot, Round 1, using Gephi 0.8.212.....	126
<b>Figure 45:</b> Synopsis of the data gathered from the ESS pilot using SOSENS .....	127
<b>Figure 46:</b> Using SOSENS data in a generic Social Network visualisation application for Social Sensing environments .....	134
<b>Figure 47:</b> Data collected from users of the NutriHeAI App .....	152

## List of Tables

---

<b>Table 1:</b> Comparison of the E/R, UML, XML & RDF/OWL Languages. ....	32
<b>Table 2:</b> Sample activities provided by users and respective steps/min values .....	98
<b>Table 3:</b> OQ_1 scores of sample activities provided by the users.....	101
<b>Table 4:</b> OQ_2 scores of sample activities provided by the users.....	102
<b>Table 5:</b> Weights used for point value estimation in the ESS pilot .....	111
<b>Table 6:</b> Activities recorded by category, and relevant statistics .....	148
<b>Table 7:</b> Exercises recorded by MET category and popular activities .....	149
<b>Table 8:</b> Reporting accuracy by evaluation metric .....	150

## List of Acronyms

---

<b>ABB</b>	Architecture Building Block	<b>SNS</b>	Social Networking Site
<b>API</b>	Application Programming Interface	<b>SOSENS</b>	SOcial SENSor
<b>bBRS</b>	binomial Beta Reputation System	<b>SOSENS DPa</b>	SOSENS Data Parser
<b>BMI</b>	Body Mass Index	<b>SOSENS DPr</b>	SOSENS Data Processor
<b>BMR</b>	Basal Metabolic Rate	<b>SOSENS SDS</b>	SOSENS Semantic DataStore
<b>BRS</b>	Beta Reputation System	<b>SOSENS SWC</b>	SOSENS Semantic Web Converter
<b>CRUD</b>	Create-Read-Update-Delete	<b>SPARQL</b>	SPARQL Protocol and RDF Query Language
<b>DQA</b>	Data Quality Assessment	<b>SSN</b>	Semantic Sensor Network
<b>ESS</b>	Exercise Social Sensing	<b>SW</b>	Semantic Web
<b>FOAF</b>	Friend-of-a-Friend	<b>SWE</b>	Sensor Web Enablement
<b>HUMINT</b>	HUMAN INTelligence	<b>UGC</b>	User-Generated Content
<b>IQ</b>	Information Quality	<b>UML</b>	Unified Modeling Language
<b>JSON</b>	JavaScript Object Notation	<b>URI</b>	Uniform Resource Identifier
<b>mBRS</b>	multinomial Beta Reputation System	<b>URL</b>	Uniform Resource Locator
<b>MET</b>	Metabolic Equivalent	<b>VGI</b>	Voluntary Geographic Information
<b>mf</b>	membership function	<b>WWW</b>	World Wide Web
<b>N3</b>	Notation 3	<b>XML</b>	Extensible Markup Language
<b>OQ</b>	Observation Quality		
<b>OWL</b>	Web Ontology Language		
<b>PDF</b>	Probability Density Function		
<b>RDF(S)</b>	Resource Description Framework (Schema)		
<b>SBB</b>	Solution Building Block		
<b>SIOC</b>	Semantically-Interlinked Online Communities		
<b>SM</b>	Social Media		
<b>SNA</b>	Social Network Analysis		



# 1 INTRODUCTION

In the early 2000s, the collective technological changes now described as Web 2.0 ushered in the era of User-Generated Content on the Web. In those joyous times, no more was the Web a read-only medium for the average user. Almost anyone could create content and post it online, a paradigm which was ultimately exaggerated in the end of the decade by the hallmark phenomenon of Social Media. Web sites such as Facebook and Twitter gave content creators similar tools for uploading their thoughts on the Web, but with the additional “promise” of an audience of peers within their social circle. This has fuelled creative tendencies to the point where the amount of content that Social Media users produce nowadays is staggering.

*“Facebook.com:  
The most popular  
(at the time of  
writing) Social  
Networking Site  
with more than 1.7  
billion monthly  
users”*

From a research perspective, after shuffling through the immense volume of Facebook posts, tweets, re-tweets, re-re-tweets and viral memes, one can very often find *humans* on social media acting implicitly as *sensors*, making *observations* about real-world events or themselves that are extremely varied; from their opinion on what the weather is like, to what is happening out of their window, to how many miles they ran today in their daily workout routine.

*“meme: an activity,  
concept, catchphrase  
or piece of media  
which spreads, often  
as mimicry, from  
person to person via  
the Internet*

In a way, these users can be treated as **Social Sensors**: a source of self-reported information that resembles sensor observations about a particular event or property and is accompanied by a wealth of social metadata, both very useful to a variety of scientific disciplines, especially when *integrated* with data from traditional hardware sensors. In many cases, social sensor information can even provide complete, feasible and economical alternatives to costly or unpractical physical instrument measurements and other forms of data-gathering, such as lengthy and unreliable questionnaire deployments, while at the same time providing crucial metadata that fuels further research.

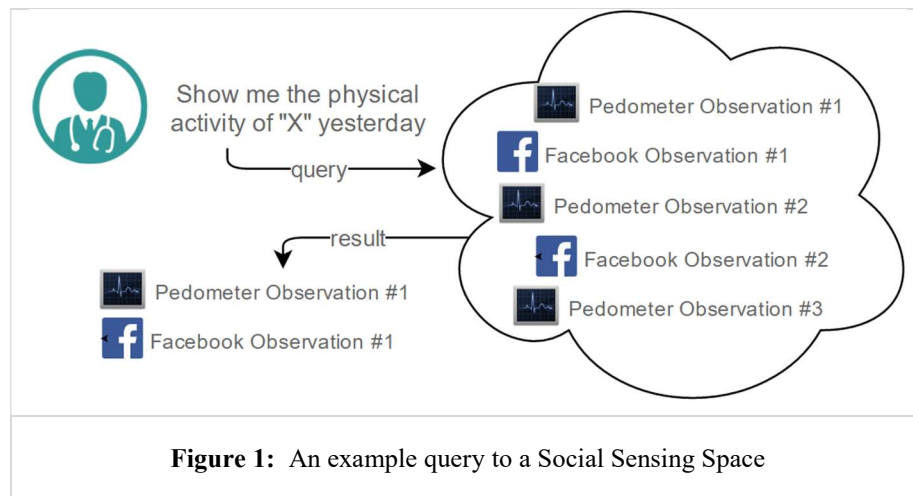
## 1.1 Motivational Scenario

### Exercise Social Sensing

Suppose that a group of people regularly post information about their exercise regimes on Facebook. These people can be treated as *Social Sensors* that observe and detect the *physical activity* property for a

specific feature of interest: *themselves*. At the same time, after being asked by a medical practitioner, these users provide access to their digital pedometer data, a device which continually tracks the wearer's steps as long as it is worn.

By treating all statements, either user-provided or pedometer-provided, as sensor observations, a complex sensing space is created where a medical practitioner *can query the sensors of this space* about the exercise activity of users: results will be returned (in a uniform way) from Facebook statements as well as the digital pedometer, as shown in **Figure 1**. Metadata will allow the medical practitioner to determine whether the information is fit-for-use.



Although this example may sound overly specific, in reality it describes a common scenario where users, acting as Social Sensors, create observations which detect or quantify real-life events (which can also be measured by hardware sensors) and then share these observations on Social Media. For example, by substituting "Facebook" for any other Social Media outlet, "physical activity" with "temperature" and "digital pedometer" with "digital temperature sensor" this can be considered a *Temperature Social Sensing* scenario.

The rich data & metadata available in such scenarios can be consumed by numerous applications, such as (i) a research initiative gathering information about user activities (e.g. in a specific age group), (ii) a computational process that can gauge the validity of each Social Sensor claim by comparing it to ground truth and use this information to calculate the credibility of each Social Sensor as a data source or

(iii) an application performing Social Network Analysis to determine how activity is correlated between peers.

## 1.2 The Research Problem

In the past years, numerous disciplines of Computer Science have dealt with how to get data “out” of Social Media. Data mining, Event/Sentiment detection, Stream monitoring & annotation as well as many other similar approaches exist for identifying and collecting data from Social Media. Researchers within numerous disciplines have used these techniques to showcase the capabilities of Social Sensing in a variety of impressive works, many of which inter-disciplinary (see Chapter 2.3.2).

Nevertheless, there has been no holistic, abstract approach to Social Sensing, resulting in troves of useful data from Social Sensing experiments, expressed in application-specific semantics, “trapped” within information silos and without the **provenance** information required to determine *fitness-for-use* and *data quality*. These issues prohibit the *exploitation* as well as the *integration* of Social Sensing information, either between social & hardware sensors or between different sensing spaces, and need to be resolved in a *systematic* manner.

**“Provenance:**  
*information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness*

As such, there is a definitive need for a *meta-modelling semantic scaffold* for Social Sensing that will allow information from this novel research space to be shared, integrated and easily consumed by various scientific disciplines. This thesis builds upon established data collection paradigms as black boxes and, rather, focuses on their results and processes and how best to describe them in order to create machine-understandable information, re-usable parts of knowledge and interoperable sensing spaces.

## 1.3 Research Questions

This thesis explores the concept of treating users as Social Sensors and proposes using **Semantic Web (SW) technologies** to bring them to the same semantic level as a Hardware Sensor, using a well-structured ontological scaffold that allows the integration of the two when needed. Semantic Web technologies, with the end-goal of making information machine-understandable, clarify a domain and set the basis for interoperability and application-independent pieces of information.

Naturally, the overall research question that permeates this thesis is:

*Can Semantic Web technologies be used to model and enhance Social Sensing spaces, acting as a “scaffold” for the integration of Social and Hardware sensors? How is that possible and what are the benefits of such an approach?*

To answer the above, this thesis explores the inter-disciplinary space of Social Sensing, drawing from existing solutions as well as providing novel approaches. In more detail, the research questions that need to be answered are:

Semantic Web	What are the “main ingredients” of the SW in its current state?
	What does a “Semantic Web” approach entail?
	What are the benefits over traditional approaches?
Social Sensing	What are the basic characteristics of a Social Sensor and how are they related to traditional hardware sensors?
	What extra metadata accompanies Social Sensor observations and how can it be modelled?
	How can Social Sensor observations, their metadata and the respective <i>provenance</i> information be modelled and expressed in ways similar to “classic” Hardware Sensors using SW technologies?
Applicability	How are SW technologies ultimately applicable in everyday practice?
	What is needed to design an implementation-independent framework for Social Sensing using SW technologies?
	How is such a framework applicable to a real-life Social Sensing scenario?

## 1.4 Research Goals & Document outline

Given the above, this thesis defines the following goals, which are explored within the relevant chapters:

**Chapter 2 (“The Research Space”):** Examine the research space of Human Sensing, Social Media as well as recent technological enablers in the Semantic Web technology stack. Explore the benefits of an approach using Semantic Web technologies and define the “basic ingredients”.

**Chapter 3 (“Modelling Social Sensing Environments”):** Enhance, restructure and represent data obtained from *humans* through Social Networks and data obtained through *sensors* in a suitable common format, using Semantic Web technologies. Emphasize the re-use of existing ontologies where possible, keeping in line with the Semantic Web vision. Explore the role of provenance in Social Sensing scenarios and set the grounds for data quality assessment.

**Chapter 4 (“The SOSENS framework”):** Explore the applicability of the above in everyday practice and design an abstract semantic framework to “scaffold” Social Sensing spaces, with the appropriate human interfaces where necessary. Determine requirements based on Chapter 3 rather than actual implementations, to support the application-independent nature of the framework.

**Chapter 5 (“Pilot Implementation”):** Assess the usefulness of the proposed framework through a full-scale pilot Social Sensing scenario, where observations about an event come from both social and hardware sensors. Provide a “demo” of the SOSENS framework in real-life conditions and describe the specificities of describing social sensors in such a case.

**Chapter 6 (“Conclusions”):** Summarize the findings of this research, evaluate its contributions to Social Sensing and highlight important future directions.



## 2 The Research Space

### 2.1 Humans as Sensors

Goodchild et al, in one of the pioneering publications of citizen science [1], distinguish *three* types of sensor networks:

1. The first, a network of *static and inert sensors* that captures specific measurements of the local environment, is a – nowadays – ubiquitous concept.
2. The second, where *sensors are carried by humans*, vehicles or animals, is also gaining traction as many projects appear that equip users with sensors that measure the users themselves or the environment around them.
3. The third, **has humans acting as sensors themselves**, *each equipped with some working subset of the five senses and with the intelligence to compile and interpret what they sense*”.

It is true that as far as cognitive capabilities go, it’ll be quite some time before the human mind as a whole is seriously threatened by something that occupies less than a small room [2], but human sensing is definitely not a new concept; humans are one of the most versatile and unique sources of information about processes and relationships that exist in their spaces [3]. A human, as opposed to a hardware sensor, may reveal information that only exists in the user’s mind and may otherwise be impossible to obtain. He/she is also capable of providing information about *future* situations or *intent* that can be very valuable and, again, is very hard to obtain via traditional sensing means. Human involvement is also particularly useful in sensing various processes in complex personal, social, and urban spaces where traditional sensor networks suffer from gaps in spatiotemporal coverage, limitations in making complex inferences, inability to adapt to dynamic and cluttered spaces, and aesthetic and ergonomic problems [3].

*In [3], Human Intelligence is matched with Watson, a closet-size IBM computer that won the popular US game Jeopardy at human expert-levels in terms of precision, confidence and speed.*

In the defense and security arena there is a long history of information gathered via HUMINT (HUMAN INTelligence) as opposed to electronic sensors. Ecological Momentary Assessments [4] of human subjects are commonly used in social and behavioural sciences to

acquire information that is hard to get from physical sensor sources. Human Sensors are also commonly treated as sources of *self-reported data* in Healthcare and have been an essential component of Health research since its inception. In Nutrition & Dietetics for example, food and physical activity diaries are one of the most important tools in gathering patient data [5], [6]. In [7], subjective methods (‘direct observations, diaries, activity logs, recall and questionnaires’) are defined as very popular methods for quantifying the selected variable (in that case, physical activity) due to their relatively low cost and the added value of contextual information provided by the user. This is especially true in large-scale studies, where cost and ease of deployment can become a very important factor in the overall success and results of the study. The concept of humans as sensors also has strong roots in the Voluntary Geographic Information (VGI) community, where users are inherently used to report geographical information such as natural events ([1],[8])

Another classification of human sensors can be seen in “Human-centric sensing”, a related research roadmap by Srivastava et al [3], where the role of humans in sensing is divided into:

- humans as targets of sensing
- humans as sensor operators
- **humans as data sources**, and
- humans as part of data processing (annotation, data triage, data analysis & fusion).

Of course these roles are not necessarily mutually exclusive and often humans are playing multiple roles in a sensing scenario. The authors also note that the *nature* and *purpose* of human sensing is also important to consider. Humans may participate in a voluntary, opportunistic, incentivized, directed or organised way and that can affect the way they contribute data. The purpose of their participation is also a factor: one can be collecting sensory information for self-analysis, or for a specific purpose (such as a top-down directed sensing campaign operated by a third party).

In [9], Sheth, taking into account the prevalence of the Web, further defines the role of such human sensors as “[*humans as*] *citizens on the ubiquitous Web, acting as sensors and sharing their observations and*



views using mobile devices and Web 2.0 services”. Using Web services (fora, blogs, social media) humans act implicitly as sensors, making observations that are extremely varied – from their opinion on what the weather is like, to what news is trending at the moment to how many miles they ran today during their daily workout routine.

*“[in Web 2.0 environments], humans act implicitly as sensors, making observations that are extremely varied – from their opinion on what the weather is like, to what news is trending at the moment to how many miles they ran today during their daily workout routine.”*

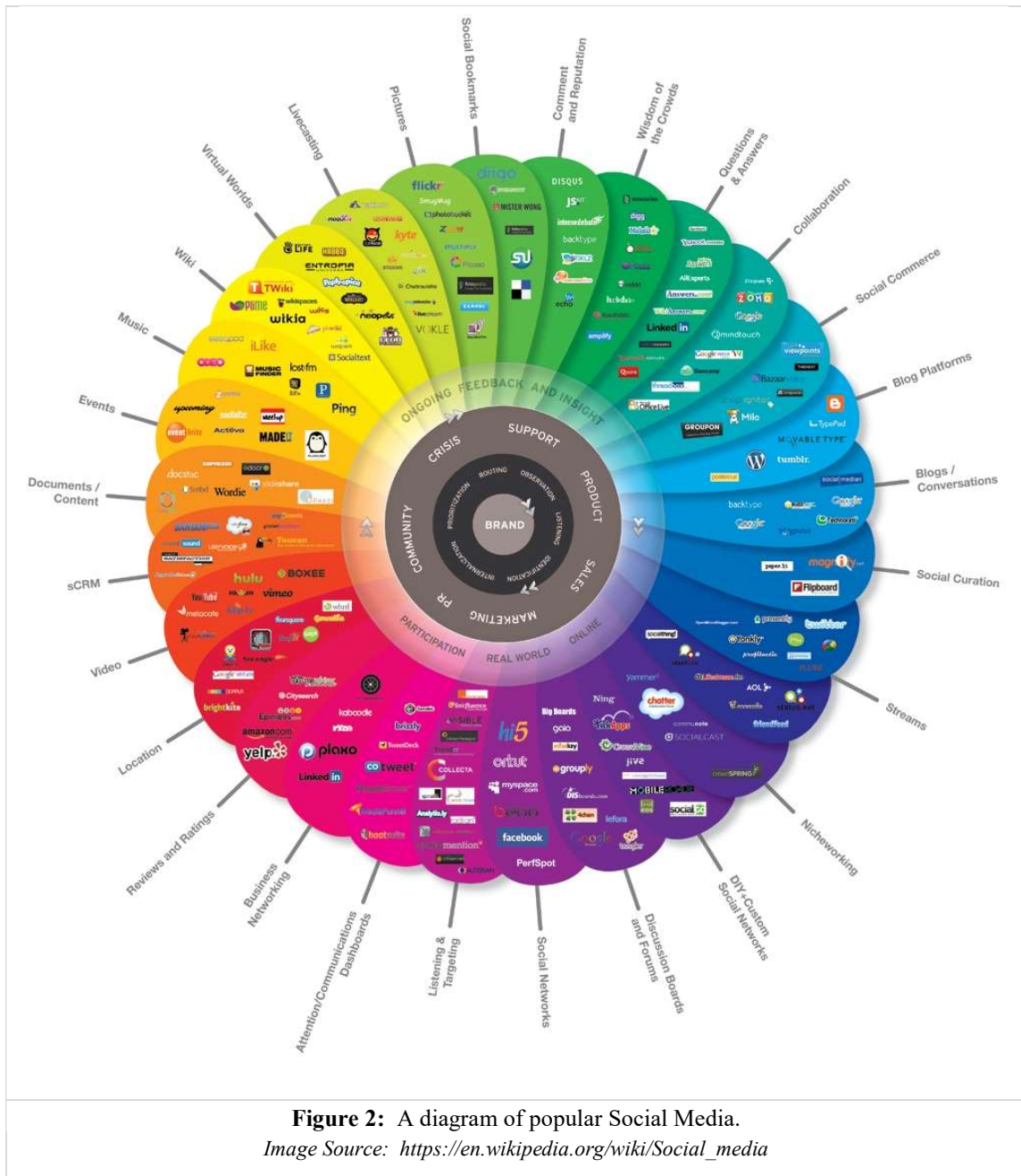
## 2.2 Social Media: A vast repository of Human Sensor Observations

Nowadays, most people have a “digital life” and produce digital data streams as well as digital footprints on various services of the World Wide Web, typically defined as Web 2.0 services because of their user-generated content. A good example of the modern evolution of this practice which begun on Fora & Blogs, is *Social Media* (SM) platforms such as Facebook and Twitter.

Social Media are computer-mediated technologies that allow users to view, create and share information, ideas, and other forms of expression via virtual communities and networks [10], [11]. While implementations vary, all SM have the following things in common:

1. They are interactive Web 2.0 Internet-based applications
2. They provide mechanisms for the creation and management of *user-generated content* such as text posts or comments, digital photos or videos, as well as data generated through all online interactions
3. SM users create service-specific *profiles* for the website or app, that are designed and maintained by the social media organization
4. SM facilitate the development of *online social networks* by connecting a user's profile with those of other individuals and/or groups

The variety of current implementations is overwhelming. **Figure 2** shows an aggregated diagram of popular SM where 29 different categories are defined, ranging from events and location to social commerce and Business networking. Some of the platforms are not only popular within their category, but also some of the most popular platforms globally.



Facebook, the current “champion” of Social Networking Sites (SNS - which can be considered a subset of Social Media) is consistently the second or third most visited website in the world [12], and is sporting a stunning *1.8 billion* monthly active users worldwide [13]. Even if Facebook is dethroned from this position, another form of SNS will quickly fill the gap, as what was once a trend of specific age groups is now a staple of the Web [14].

In Social Media, users act *naturally* as data sources by volunteering information they care about, be it for themselves or for other platform users. While users are acting without any prior coordination and without downloading any specialized sensing application, their “output” can be treated as the output of a *human sensor*. Not only that, but information on Social Media is typically *augmented by social metadata* (friends, connections, group memberships etc.) that can play multiple roles; establish context and/or power social network analytics, aid a researcher in reaching conclusions about the user and his/her environment etc.

A. Karimi et al.:  
“My grandma uses  
Facebook” [15]

As such, Social Media can be treated as a vast *repository* of unstructured human sensor observations, bundled with *valuable* social metadata, where the sensor is the human entity that operates the Social Media account. These observations are, subsequently, available to be exploited and consumed by researchers who place high-value in human observations.

“Social Media can be  
treated as a vast  
repository of  
unstructured human  
sensor observations,  
bundled with valuable  
social metadata “

## 2.3 Social Sensing

In recent research efforts, the term *Social Sensing* has prevailed, characterising Social Media users as *Social Sensors* ([15],[16],[17]). In their book about Social Sensing, Wang et al [18], define the term as referring to three types of data collection:

1. *Participatory sensing*, where individuals are explicitly and actively involved in the sensing process, and perform critical operations such as operating the sensors
2. *Opportunistic sensing*, where individuals are passively involved, for example, by pre-authorizing their sensing device to share information on behalf of the owner
3. *Social data scavenging*, which refers to a sensing paradigm, where individuals remain unaware of the data collection process. An example is where social networks are treated as sensor networks. Public data posted on social networks (e.g., Twitter) are searched for relevant items. In social data scavenging, the participants “agree” to the fact that their posts are in the public domain and they are simply unaware how the public may actually use their information.

*“The nuances of traditional hardware sensing are different than the ones of Social Sensing and should not be mixed”*

Arguably, this is a very broad definition which intentionally “re-includes” traditional hardware sensors in the loop, mainly because of their prevalence and ubiquity in early data-collection initiatives such as CarTel [19], CabSense [20] and BikeNet[21]. Ideally, Social Sensing should be defined *only by* the equivalent of “social data scavenging”, as it is the definition which more closely resembles the *novel* and *unique* features of this paradigm. The nuances of traditional hardware sensing (be it in participatory or opportunistic sensing) are different than the ones of Social Sensing and should not be mixed, even if the data dissemination by hardware sensors takes place in a social-network-like environment.

Thus, for the purposes of this work, the term *Social Sensor* is used as an ‘umbrella term’ to represent the observations made by *Human Sensors* in Social Media such as a Social Networking Site.

Social Sensors: Humans that act as sensors, sharing their observations on Social Media.

For example:

- In a scenario where users are talking about their *physical activity* in *Facebook posts*, they are acting as *activity sensors* in the sense that they self-report the activity performed by them, in the same way a “traditional” activity detection sensor would.
- When they report how the weather feels like in a geo-tagged Twitter post, they are acting as *temperature sensors*.

In both cases, they are acting as *Social Sensors* because they disseminated this information in Social Media.

### 2.3.1 Combining Hardware & Social Sensor Observations

The accepted definition of Social Sensing should not be mistaken as taking hardware sensors “entirely out of the loop” – it just differentiates them from Social Sensors. The *integration* of hardware & social sensing in the same environment is still of great interest in

this thesis, as in many Social Sensing environments, humans are not the only ones capable of monitoring a specific event or property. More often than not, the target of sensing can be concurrently monitored (in the same time or in the past) by a “traditional” hardware sensor. Resch [22] defines People as Sensors as:

*[People as Sensors is] “a measurement model, in which measurements are **not only** taken by calibrated hardware sensors, but in which **also humans can contribute** their individual ‘measurements’ such as their subjective sensations, current perceptions or personal observations”*

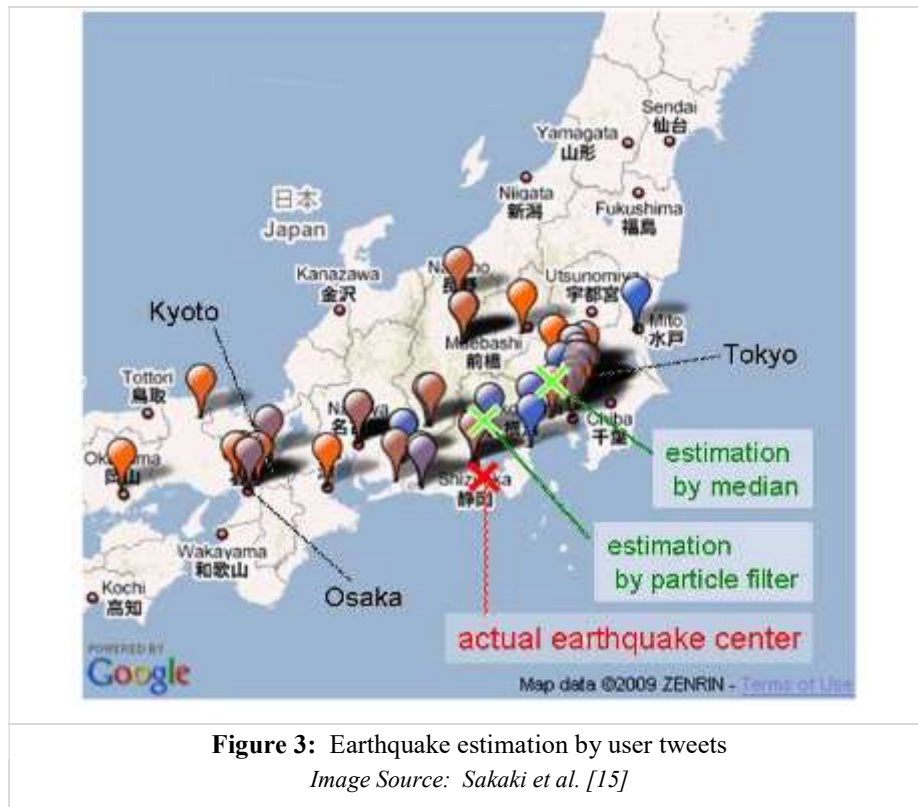
The benefits from systematically coupling human observations with “traditional” hardware sensors are numerous. First of all, it allows a user on the monitoring end to perform queries that are source-agnostic and thus, more to the point. For example, a search for “physical activity observations for person A” could return an observation of “2 hours of dancing on Monday [via Facebook]” and “1.5 hours of dancing on Tuesday [via Worn Sensor]”. When such complex queries become transparent, it is a major facilitator for disciplines such as health professionals where added burden (overhead) is an important issue [23].

In addition, the ability to compare the *perceived* by a human vs. the *perceived* by a sensor observation can be very useful, for example in cases where the sensor is treated as ground truth. Drawing again from the motivational scenario example regarding physical activity monitoring, this can be the user’s perceived duration of an event (“I ran 45 minutes”) compared to the sensor’s more accurate reading (“activity duration: 30 minutes”). If done in an efficient and systematic way, it is possible to provide applications and the researchers using them with much higher degrees of context-awareness, further shortening the gap between digital and physical worlds and overall providing a much better service [16].

Finally, the existence of both types of sensors monitoring the same property can provide an effective solution to the problem of information continuity; suppose a scenario where a property cannot be continuously monitored by a hardware sensor, due to power needs or coverage capabilities. In this cases, the Social Sensor can “fill the gap” by providing information about the event.

### 2.3.2 Examples of Social Sensing in Research Literature

Social Sensing is a very “young” term which can be traced to the seminal paper by Sakaki et al [15]. In this work, a system “detects” earthquakes by treating Twitter users as sensors and applying Kalman filtering and particle filtering for event detection. In the end, the distance of the socially-sensed event to the hardware-sensed event is calculated, with very good result rates (see **Figure 3**).



**Figure 3:** Earthquake estimation by user tweets  
*Image Source: Sakaki et al. [15]*

Sakaki’s work, now cited over 2500 times, can be considered the pioneering work in Social Sensing, as it validated what many researchers and users of Social Media had been speculating: people are eager to report on things that interest them or may interest their social circle and, with appropriate techniques, they can be used as Social Sensors.

Other researchers have also used Social Sensing for detecting similar natural hazard events (earthquakes, floods etc.) with rates of detection that are comparable to or, in some cases, even faster than detections by equivalent physical instruments [24]–[26]. Twitter users have even

been used as real-time sensors of sports games [27], as sensors for detecting downtime on internet services [28] and even as sensors for predicting stock market trends [29].

Wikipedia & SciStarter maintain a link of over 1100 active citizen science projects [30] such as BeeID, where members of the public are encouraged to upload photos of bees and "geotag" them to place them on a map. With the advent of Social Media, many of these projects have "moved" to using them as a platform of communicating and disseminating user-aggregating content (e.g. BeeID used Flickr).

In Healthcare, researchers have used data from Facebook & Twitter in many novel ways, such as early detection & identification of contagious outbreaks [17], [31], [32]. In [33], the authors track the H1N1 flu pandemic in the UK using twitter, with success rates as high as 95% (i.e. the results of the Twitter Social Sensors correlate with data from the UK Health Protection Agency). Since health is often a "socially discussed" topic, many e-Health related communities (or groups) can exist within Social Media. For example, the PatientsLikeMe.com Social Networking Site is a repository of patient-generated data about illnesses & the treatments followed by patients. The "1000 steps" Facebook group (now closed) was a group where users equipped with pedometers post their "steps" to share and compare with similar-minded (and similarly-equipped) users.

In the EU project research space, the SUPER FP7 research project [34] uses a NoSQL meta-model to explore the use of social media in emergencies and security incidents. The Social Sensing.it project developed a similar framework for Early Warning Emergency Management. Finally, the Social Sensor project [35] collects, processes, and aggregates big streams of social media data and multimedia to discover trends, events, influencers, and interesting media content.

## 2.4 (re-)Defining the problem

The above is just an indicative subset of the possible uses of Social Sensing which may be a young scientific research area but, given the rapid popularity of Social Media platforms such as Facebook and Twitter, will become even more important in the coming years.

However, as numerous researchers have pointed out ([16], [22], [36]), there is a definitive need for *common* and *re-usable* ways to describe the Social Sensing process. A recurring issue is that data from social sensors is mostly unstructured, untrusted and usually provided without the *provenance chains* that are required for assessing data quality and, eventually, decision making. Even if data-modelling methodologies are used, data from Social Sensing spaces is usually application-specific and kept in an information silo.

“The goal of this thesis is not to **contradict** but to **complement** such research by working on the data modelling level”

The goal of this thesis is *not to contradict* **but to complement** social sensing research by working on the data modelling level, in order to create a semantic “scaffold” for heterogeneous sensing environments that is *implementation-agnostic*. Of special importance is finding a uniform way to describe the knowledge relevant to Social Sensing data creation, ownership and transformation which can be thought of as the ‘life-cycle’ of such environments. As noted before, this thesis does not deal with event detection or social data mining, but rather with their *results and processes* and how best to describe them in order to create re-usable parts of knowledge and interoperable sensing spaces & experiments.

To visualise a very primal part of the problem and its intended solution, consider an example post of a Social Sensor on Facebook which reports on an earthquake as shown in **Figure 4**.



**Figure 4:** An example Social Sensor post on Facebook



What this post really contains, after being extracted from the Social Media Platform (e.g. here: using the Facebook API [37]) is:

- The user who created the post, along with his social circle and social media profile (details & metadata of the user)
- The Facebook post itself, along with when it was created, how it was shared etc. (details & metadata of the user's post)
- The resulting Social Sensor observation; when, where and what happened (details & metadata of the observation)
- How this observation was created & captured (details & metadata of the observation' provenance in relation to the Facebook post)

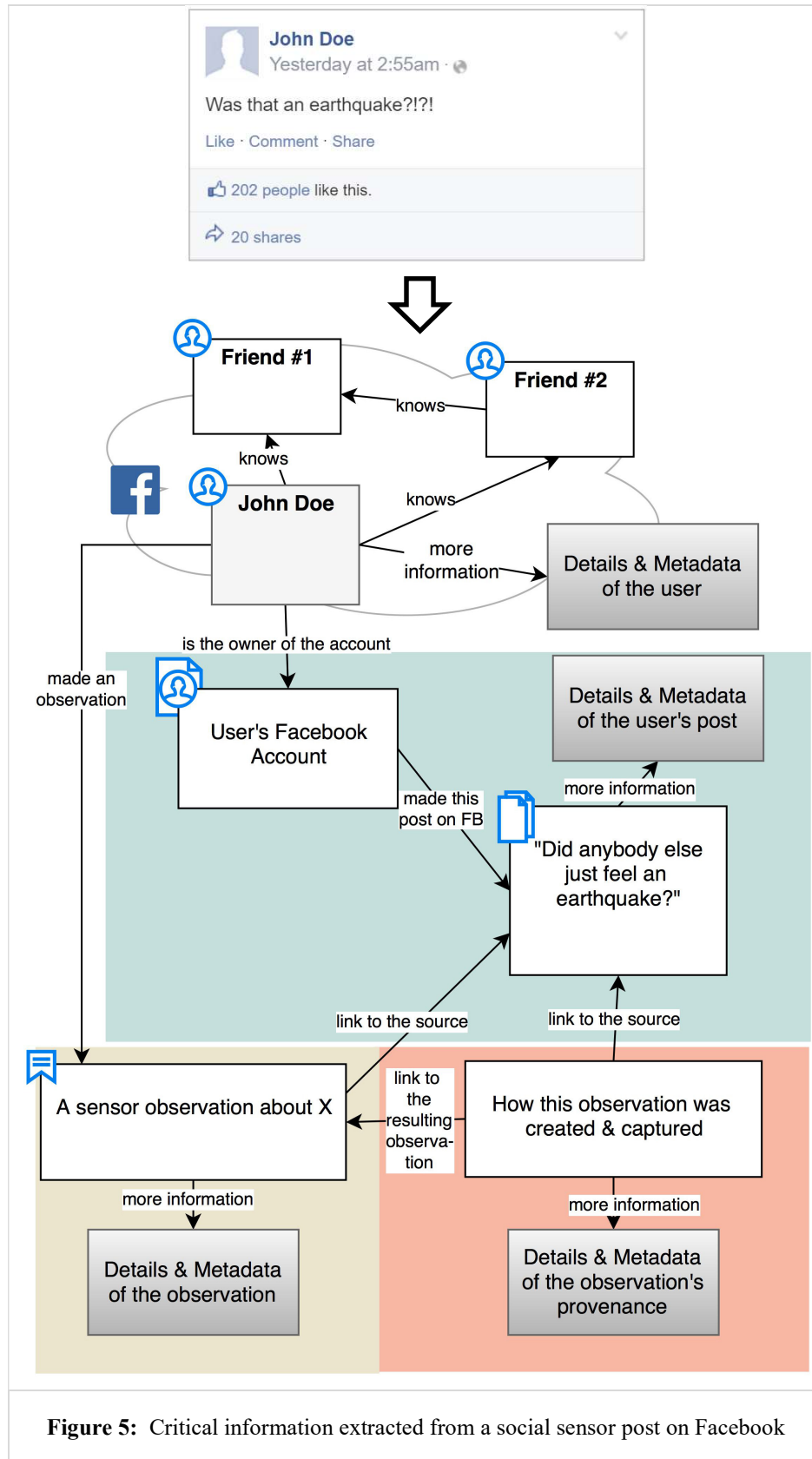
This information is critical because (a) it showcases the augmented metadata that accompany each Social Sensor observation and (b) can bring a Social Sensor to the same semantic level as a Hardware Sensor, allowing the integration of the two when needed, as can be seen in the next Chapter. In essence, each observation can be broken down to the information above, as visualised in **Figure 5**.

*“[this metadata] can bring a Social Sensor to the same semantic level as a Hardware Sensor “*

The end goal of this thesis is to find a way to define and track this important (meta)data. To achieve this purpose, it leverages *Semantic Web* technologies; a set of technologies that ultimately clarify a domain's structure of knowledge and enable knowledge-sharing in an effective way. The basic aspects of the Semantic Web, as required for the purposes of this thesis, are discussed in the chapter that follows.

## 2.5 THE SEMANTIC WEB

The Semantic Web (SW) is about making the Web and the applications that use it smarter. There is a staggering amount of data available on the Web, but it is mostly unstructured and many times it resides in “information silos”, databases within websites which cannot be accessed outside of that website's context.



**Figure 5:** Critical information extracted from a social sensor post on Facebook

For example, in Social Media, why doesn't updating your Facebook profile also update your LinkedIn profile (i.e. "the walled garden" effect, see **Figure 6**)? Why do you need third-party solutions to "post" to many Social Media at the same time? This is an issue with most of the information online at the moment – it is siloed and, above all, it is not structured well enough to be understandable by humans and machines alike.

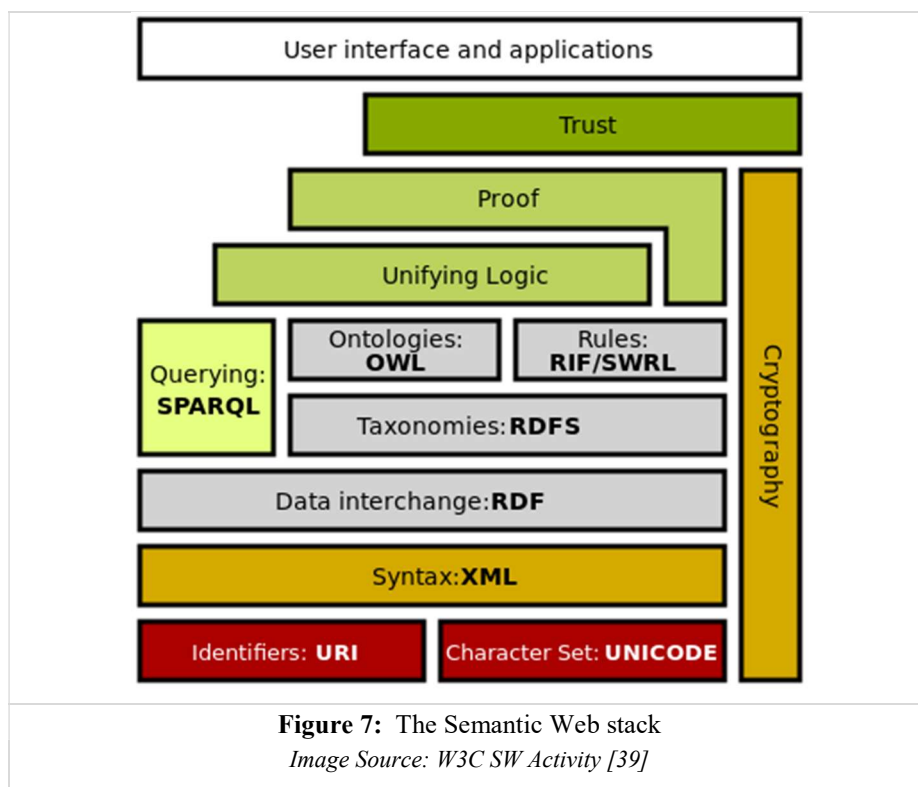


**Figure 6:** The "walled gardens" of the early 2000s era of Social Networking.  
 Image Source: Yeung et al (2009) & the Economist (2008 print issue) [40], [41]

The Semantic Web movement began around the early 2000s with Sir Tim-Berners Lee, the inventor of the WWW presenting his idea of a "Web of Data" [38]. The frontpage of the resulting W3C Semantic Web Activity defines the SW being about two things [39]:

- (1) It is about *common formats* for integration and combination of data drawn from diverse sources.
- (2) It is also about *language* for recording *how the data relates to real world objects*. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing."

The W3C Semantic Web Activity was subsumed in 2013 by the W3C Data activity [42] but the basics remain the same. A “new” Web must be built upon the now-established Web 2.0 which gives meaning to the petabytes & exabytes of data available within its services. In this new Web (which many call Web 3.0), a web user, be it a human or machine, should be able to seamlessly navigate between data sources on the Web. This is made possible by a suite of (now) well-established conventions, languages & technologies, commonly referred to as the “Semantic Web stack”, as shown in **Figure 7**): Syntax (XML), Data Interchange (RDF), Ontologies (OWL) and Querying (SPARQL).



The basics of the Semantic Web, as needed for this thesis, are presented in the next section starting from its first unique block: RDF; more information can always be found at the canonical citation by Tim Berners-Lee [38] or online [43].

### 2.5.1 Data Interchange: RDF

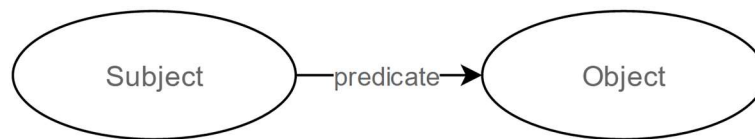
*RDF*, originally created in early 1999 by W3C, stands for Resource Description Framework, and is a standard for encoding metadata. Its goal is to define a mechanism for describing *resources* that makes no

assumptions about a particular application domain (domain independent), and therefore can be used to describe information about any domain.

Each *resource* is identified using a *Uniform Resource Identifier*, or *URI*. URIs are simply Web identifiers, like the strings starting with "http:" or "ftp:" that are encountered on the World Wide Web. Anyone can create a URI, and the ownership of them is clearly delegated, so they form an ideal base technology with which to build a global Web on top of. In fact, the World Wide Web is such a thing: anything that has a URI is considered to be "on the Web".

It is possible to make *statements* in RDF about these URIs. Statements are represented as a *triple*, which has (see **Figure 8**):

- a Subject (S)
- a Predicate (P) and
- an Object (O).



**Figure 8:** Statements in RDF

This is not really much different than statements made in natural language. For example, two very basic *triples*, in natural language are:

Subject	Predicate	Object
Ion Pagkalos	is a	Person
This thesis	is written by	Ion Pagkalos

To express this in RDF, each of these 5 resources (Ion Pagkalos, is a, Person, is written by, this thesis) needs to be assigned a URI, the “base” of which for now can just be <http://example.com/#>, so that for example, the unique URI of the “Ion Pagkalos” resource is

`<http://example.com/#Ion_Pagkalos>`

RDF, similarly to XML, allows one to define namespaces in order to skip re-writing the base part of the URI, for example `e:` can “stand for” the `example.com` namespace:

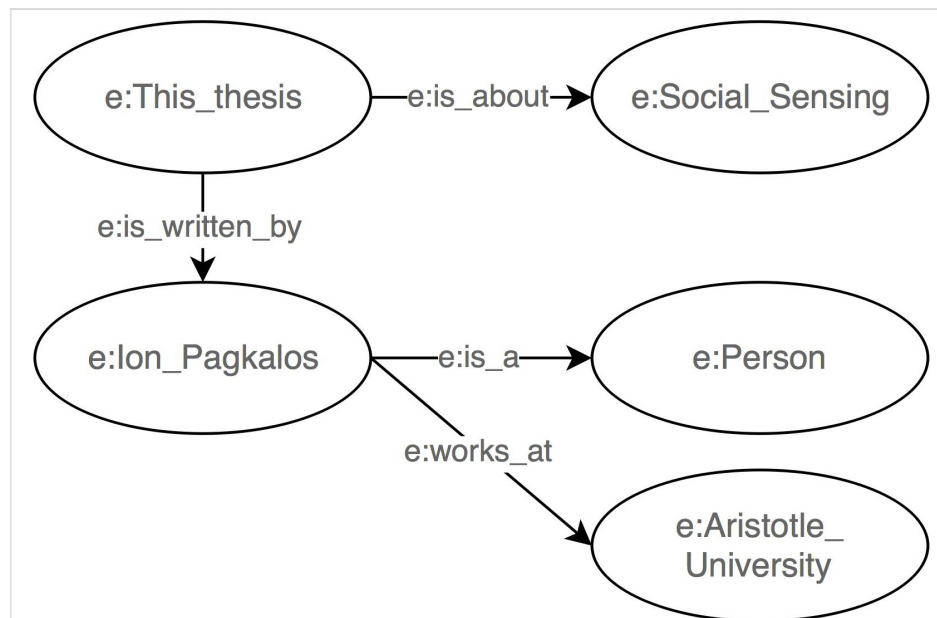
```
@prefix e : <http://example.com/#>.
```

Subject	Predicate	Object
e:Ion_Pagkalos	e:is_a	e:Person
e:this_thesis	e:is_written_by	e:Ion_Pagkalos

RDF has the added benefit of being flexible enough to express any information anyone can think of about either the author, or this thesis, by just “adding rows” to the table above, for example:

e:Ion_Pagkalos	e:works_at	e:Aristotle_University
e:this_thesis	e:is_about	e:Social_Sensing

These rows are exactly equivalent to a directed graph representing a collection of statements, called an *RDF Graph*:



**Figure 9:** An example RDF Graph

This abstract model can represent any knowledge as long as that given knowledge can be expressed as a labelled and directed graph [44] (such as the one as shown in **Figure 9**). Any new fact (statement) can be easily added to an existing graph to make it more expressive, and, without any loss to its original meaning, any such graph can be expressed as a collection of RDF statements, representing a concrete implementation of the RDF abstract model.

RDF can be expressed using multiple *serializations*; RDF/XML[45] is considered to be the standard interchange format for RDF on the Semantic Web, although other serializations, such as Notation3 [46] and its subset, Turtle [47] are considered more user-friendly and human-readable. For example, the information of **Figure 9** in Turtle can be written as:

<pre>e:Ion_Pagkalos   e:is_a e:Person ;   e:works_at e:Aristotle_university .  e:This_thesis   e:is_about e:Social_Sensing ;   e:is_written_by e:Ion_Pagkalos .</pre>	RDF/Turtle
---	------------

In this specific example, all Objects (or property values) are other, unique RDF resources. It is also possible to use simple raw text data as objects, called RDF Literals. In RDF/Turtle, to use literal values, the value can be enclosed in double quote marks and can be optionally localized by attaching a language tag such as:

<pre>e:Ion_Pagkalos e:first_name "Ion"@en .</pre>	RDF/Turtle
---	------------

Finally, RDF also supports “blank nodes” (or anonymous nodes) which exist for the cases where properties need to be “attached” to subjects/objects for which you don’t want to define a URI. This can simply be written as:

<pre>_:a1 e:first_name "Ion"@en .</pre>	RDF/Turtle
---	------------

The above can be roughly translated as “there’s someone who’s called Ion”. In reality, this blank node will most likely be assigned a local identifier so that it could be referred within the same document scope, but it is a very useful convention for resources that need not be

referenced from “outside” the RDF graph or for transient data that’s not meant to be stored.

## 2.5.2 Ontologies & Taxonomies: RDFS & OWL

RDF is a flexible, abstract model for making statements about resources, but it lacks the ability to define the *structure*, & *vocabulary* used in those statements. RDF Schema (RDFS [48]) is a W3C standard for describing these aspects, using a common language and a defined, shared vocabulary. RDFS defines classes, properties, and the relations between them that are allowed to appear in an RDF specification and RDF is used to specify the content (i.e., the actual information).

For example, we can make `e:Person` an `rdfs:Class` and `e:Ion_Pagkalos` a member of this class as such<sup>1</sup>:

```
e:Person rdfs:type rdfs:Class .
e:Ion_Pagkalos rdfs:type e:Person .
```

RDF/Turtle

All resources who have `e:Person` as value for their `rdfs:type` property belong to the RDFS class `e:Person` and are considered *instances* of this class. A resource can be an instance of several classes, and a class can also have subclasses, e.g.:

```
e:Researcher rdfs:subClassOf e:Person
```

RDF/Turtle

Instances of a subclass are also members of classes higher in the hierarchy (i.e. if `Ion_Pagkalos` is a `Researcher`, then he’s also a `Person`), which sets the base for concept hierarchy. In a similar way, we can also define some of the other resources as `rdfs:Property`:

```
e:works_at rdfs:type rdfs:Property .
e:is_about rdfs:type rdfs:Property .
e:is_written_by rdfs:type rdfs:Property .
```

RDF/Turtle

The `rdfs:range` & `rdfs:domain` properties can be used to further

---

<sup>1</sup> In RDF/Turtle, you can also use the shorthand for `rdfs:type` by just writing “a”, e.g. `e:Ion_Pagkalos a e:Person`.



define which classes the subject and object of each property belong to. For example, we can declare that when the `e:is_written_by` property is used, the subject is something of the `Book` class, and the object is a `Person`<sup>2</sup>:

<code>e:is_written_by</code> <code>rdf:domain</code> <code>e:Book</code> .	<i>RDF/Turtle</i>
<code>e:is_written_by</code> <code>rdf:range</code> <code>e:Person</code> .	

Similarly to `rdfs:Class`, properties also can define a hierarchy by using `rdfs:subPropertyOf`. More information about the entirety of the RDFS spec's features can be found in the specification, online [48].

OWL (Web Ontology Language [49]) *extends* RDFS and provides the means to describe *more complex relationships* between classes and their properties, allowing the design of *domain ontologies*. **Ontologies** can be thought of as the formulation of an exhaustive and rigorous conceptual schema in a given domain [50]. OWL is not confined to the Web, and it has been applied successfully for knowledge modelling in many application areas [51]. Modelling information in OWL has two practical benefits: as a descriptive language, it can be used to express expert knowledge in a formal way, and as a logical language, it can be used to draw conclusions from this knowledge.

OWL is built upon RDF Schema and, as such, all the terms contained in the RDFS vocabulary can be used when creating OWL documents. However, OWL expands upon RDFS with new conventions such as:

- Describing data in terms of set operations (e.g. unions of classes). For example, defining the `Father` class as the union of people that belong to the `Parent` & `Man` classes:

```
e:Father owl:unionOf (e:Parent, e:Man)
```

*“Ontologies can be thought of as the formulation of an exhaustive and rigorous conceptual schema in a given domain”*

---

<sup>2</sup> A common misconception when developing SW applications is that these properties are ‘enforced’ in some way. These properties do not prohibit a resource to be declared as the subject or object of the property; they just make it clear that, if it is either of the two, it “automatically” is classified as the domain or range respectively.

- Describing equivalences across databases with `sameAs`. For example, the same person across two different databases:

```
DB1:Person1 owl:sameAs DB2:Person2
```

- Describing additional property characteristics, such as stating a property to be the inverse of another property using `inverseOf`:

```
e:is_written_by owl:inverseOf e:is_the_author_of
```

- OWL restrictions: Restricting the value that properties can take with `allValuesFrom` and `someValuesFrom` and the number of property values that a class member can hold for a given property (cardinality, `minCardinality` and `maxCardinality`). For example, a “committee” on which all members must be persons and must have at least 2 female members:

```
:SelectionCommittee
  a owl:Class ;
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:onProperty :committeeMember ;
      owl:allValuesFrom :Person
    ] .
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:onProperty :committeeMemberFemale ;
      owl:minCardinality "2"^^xsd:int
    ] .
```

OWL comes in different flavours (or profiles) which trade expressive modelling power for computational efficiency when performing reasoning (see the next section). OWL 1, the first version of the spec, has the OWL Full, OWL DL and OWL Lite flavours while the latest version of the OWL Spec, OWL 2 [52] provides the following:

- OWL2 Full, the most expressive but also the most computationally-intensive profile
- OWL2 EL, useful in applications employing ontologies that contain very large numbers of properties and/or classes.

- OWL2 QL, aimed at applications that use very large volumes of instance data, and where query answering is the most important reasoning task.
- OWL2 RL, aimed at applications that require scalable reasoning without sacrificing too much expressive power.

The respective W3C Recommendation, OWL2 Profiles [53] provides a thorough insight on the intricacies of each flavour. In general, the choice of flavour comes down to application requirements, with each flavour supporting a specific subset of the OWL2 Full capabilities.

Ontologies can be created, published online and re-used, which is a major component of the Semantic Web. For example, instead of having our own concept of the “Person” class, we can use the FOAF ontology [54] and its `foaf:Person` class. It should be noted that since OWL2 is a relatively new spec, most of the major ontologies available online are in OWL DL. However, OWL 2 is backward-compatible to OWL 1, which means that OWL 1 expressed in RDF syntax remains valid OWL 2.

### 2.5.3 Reasoner

The vision of the SW consists of not only having access to structured collections of information (RDF) and sets of inference rules (OWL) but also using these to *conduct automated reasoning* [38]. In knowledge management systems, a reasoner or inference engine is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with.

Reasoners provide different services, such as *subsumption testing* [55]: testing whether or not one class is a subclass of another class. They can also infer *disjointness* and *equivalence* of classes. By performing such tests on the classes of an ontology, it is possible for a reasoner to compute the inferred ontology class hierarchy, and based on given facts and rules, infer new facts. For example, if `Ion_Pagkalos` is a `Researcher` and `Researcher` is a subclass of `Person`, the reasoner will create a new *inferred* fact – that `Ion_Pagkalos` is a `Person`.

The reasoner can also determine class membership for individuals (instances) based on their properties as defined by OWL constructs.

This means that class membership does not always have to be asserted, but can be inferred by the reasoner. For example, if Researchers are people whose job is Research:

<pre>e:Researcher a owl:Class ;   rdfs:subClassOf e:Person ;   owl:equivalentClass [     a owl:Restriction ;     owl:hasValue e:Research ;     owl:onProperty e:has_Job ; ] .</pre>	RDF/Turtle
---	------------

and Ion Pagkalos performs Research:

<pre>e:Ion_Pagkalos a e:Person ;   e:has_Job e:Research .</pre>	RDF/Turtle
---	------------

then he is *automatically inferred* to be a Researcher.

Given its capabilities, a reasoner has multiple uses in a SW environment such as truth maintenance, belief revision, information consistency as well as *information creation* [56]. There are many implementations of SW reasoners, such as OWLIM, Fact++, Hermit, Pellet, Jena which are either available as standalone reasoners or packaged with Full SW IDEs/Frameworks. A comprehensive list can be found online at the W3Cs OWL/Implementations page [57].

## 2.5.4 Querying: SPARQL

SPARQL (pronounced “sparkle”) is an RDF query language and is the de-facto data access protocol for the Semantic Web. The SPARQL Recommendation [58] consists of a query language, a XML format in which query results will be returned, and a protocol of submitting a query to a query processor service remotely.

The query language can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. It also provides a protocol that can query a remote RDF data set.

SPARQL specifies four different query variations for different purposes:

- `SELECT` query, to extract raw values from a SPARQL endpoint (results are returned in a table format).
- `CONSTRUCT` query, to extract information from the SPARQL endpoint (results are returned in valid RDF).
- `ASK` query, to provide a simple True/False result
- `DESCRIBE` query, to extract an RDF graph from the SPARQL endpoint, the content of which is left to the endpoint to decide based on what the maintainer deems as useful information.

Each of these query forms takes a `WHERE` block to restrict the query (optional in the `DESCRIBE` query).

SPARQL is designed to match the data queried to a set of triple patterns called a basic graph pattern, which are similar to RDF triples, but can contain *variables* as subject, predicate, or object. Variables are indicated by a "?" or "\$" prefix. Similarly to Turtle/N3, SPARQL allows the definition of prefixes and base URIs.

For example, a SPARQL query to return the names of all researchers within a specific RDF graph is:

```
PREFIX e: <http://example.com/#>
SELECT ?person ?name
WHERE {
  ?person a e:Researcher ;
  ?person e:first_name ?name .
```

SPARQL

And the result, in tabular form would be:

Person	Name
<http://example.com/#Ion_Pagkalos>	"Ion"@en

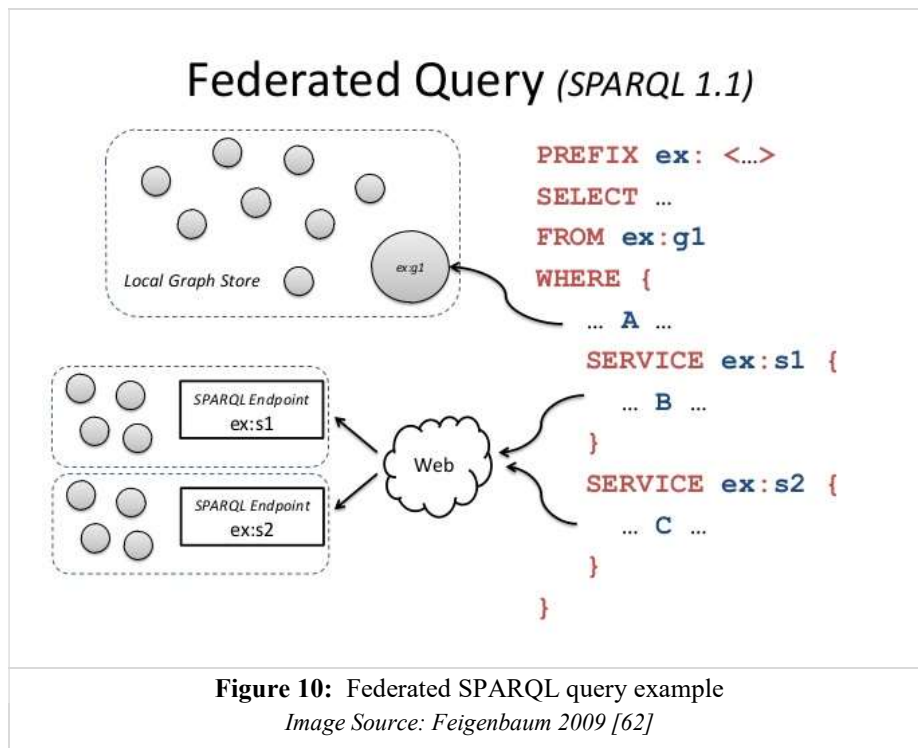
As is expected from a feature-complete query language, SPARQL supports “Solution Sequences and modifiers” [59] such as `FILTER`, `ORDER BY`, `DISTINCT`, `OFFSET` and `LIMIT`, some of which are very similar to their SQL counterparts.

The latest version of SPARQL (1.1, a 2013 W3C recommendation [60] ) adds some much-needed features to the language by specifying (among others):

- SPARQL *Update*, a language to update, create, and remove RDF graphs in a Graph Store and
- SPARQL *Federated Query extension*, for executing queries distributed over different SPARQL endpoints. More specifically, the SERVICE keyword extends SPARQL 1.1 to support queries that merge data distributed across the Web (see **Figure 10**).

*"If we view the Semantic Web as a global collection of databases, SPARQL can make the collection look like one big database"*

SPARQL is to the Semantic Web what SQL is to relational databases, but more importantly, it is a basic ingredient in realising the SW vision. Quoting of the contributors of the SPARQL working group [61]: *"If we view the Semantic Web as a global collection of databases, SPARQL can make the collection look like one big database"*.



### 2.5.5 Benefits of the Semantic Web over traditional techniques

From its initial proposal by Tim Berners-Lee, to the latest trends and initiatives, such as Linked Data [63] and DBpedia [64], the SW has progressively changed the landscape of the Web through the use and adoption of the different semantic technologies that have come along with it [65]. Massive organizations—such as Merck, Johnson & Johnson, Chevron, Staples, GE, the US Department of Defense, NASA, and others—now rely on Semantic Web technologies to run critical daily operations [66]. With SPARQL & the SW, people can focus on what they want to know rather than on the database technology or data format used behind the scenes to store the data.

Admittedly, the SW represents a paradigm shift from the traditional techniques ubiquitously used in today's information systems, such as relational databases. However, there is a trend towards moving away from relational databases where applicable, as made popular not only by the Semantic Web tech stack but also by the NoSQL family of databases<sup>3</sup>. This trend is rooted in the idea that while a relational database enables fast & efficient querying of data because of keys, indexes, tables & joins, it was never designed to accommodate *information sharing*, the *heterogeneity* of information sources, or *reasoning*, areas where the SW excels.

This is not to say that SW technologies should overtake RDBs; approaches such as RDB2RDF [67] can shorten the technology gap, combining the expressivity of the SW with the efficiency & precision of traditional RDBs.

RDF/OWL are different in their purpose from other well-known knowledge representation formalisms used in Computer Science but share many primitives with E/R, UML and XML models, as discussed by Mika in [68]. As can be seen in Table 1, where Mika compares the aforementioned formalisms, RDF/OWL is a variably expressive language which supports Distributed Representation & Formal Semantics, *effectively combining the benefits of all the approaches*.

---

<sup>3</sup> See: <https://en.wikipedia.org/wiki/NoSQL>

	Application Domain	Expressivity	Distributed representation	Formal Semantics
<b>E/R</b>	Relational Databases	*	No	No
<b>UML</b>	OO Software	**	No	Yes (in 2.0)
<b>XML</b>	Text markup and data exchange	**	Yes	No
<b>RDF/OWL</b>	Resource markup and data exchange	* to ***	Yes	Yes

**Table 1:** Comparison of the E/R, UML, XML & RDF/OWL Languages.  
Source: Mika, 2008 [68]

### 2.5.6 The Semantic Web for (Social) Sensing

A comprehensive answer to when ontologies & the SW are relevant to the problem at hand comes from Richard Cyganiak, a prime SW researcher and co-author on DBpedia publications<sup>4</sup>. He advocates that SW technologies should be used when there is a need for (i) integrating data from different sources without custom programming, (ii) offering data for re-use by other parties and (iii) decentralizing data in a way that no single party "owns" all the data. This can be done "on top of the RDF data model", which has the advantage of *not being tied to a proprietary data storage/representation technology, like a database dialect*.

As was discussed earlier, all of the above are applicable in Social Sensing. Primarily, SW technologies can assist in managing, querying, and combining sensors and observation data, thus allowing users to operate at abstraction levels above the technical details of format and integration [70]. Semantic Web technologies can bring clarity to a field dominated by heterogeneity, by making sure that social sensors and their data are represented in a uniform way.

---

<sup>4</sup> The following statements are from the W3C SW Interests mailing list [69]



In addition, machine-interpretable semantics allows autonomous or semi-autonomous agents to assist in collecting, processing, reasoning about, and acting on (social) sensors and their observations. To quote the authors of the Semantic Sensor Network ontology [70] (presented later within this thesis):

*“Shared semantic definitions help not only with data integration from multiple sources, but can also assist in integrating new data into historical, temporal and spatial contexts. Definitions of sensors and their capabilities are also useful for provenance and quality reasoning.”*

## Chapter Conclusions

This chapter presented the research spaces of Humans as Sensors, Social Media, The Semantic Web, and the emerging field of Social Sensing. As discussed, due to the inherent heterogeneity and complexity of Social Sensing spaces, there is a definitive need for organising the information in non-siloed solutions, with clear provenance and actor/process attribution metadata. Up to now, no published scientific work has attempted to tackle the problem in its entirety, using Semantic Web technologies.

The Semantic Web, as a technology stack, not only comes with a lot of benefits, but has also reached a point of maturity & stability of standards and technological solutions that makes it worthwhile (and realistic) to design frameworks to enhance Social Sensing, such as the one presented within this thesis. The chapter that follows discusses the ontological base of the framework and the nuances of describing social sensors in the same way as “classic” hardware sensors.



## 3 Modelling Social Sensing Environments

### Chapter Introduction

As discussed in the previous chapter, sensing has evolved into a multi-disciplinary scientific domain with a variety of open issues. Social Sensing, in particular, is a young scientific area where Semantic Web technologies can play a crucial role.

This chapter begins to tackle the data modelling aspect of Social Sensing environments by discussing the nature and characteristics of classic Hardware sensors. “Taking a page out of the hardware sensor playbook”, it then explores how Social Sensors can be represented in the same way, in order to express them in a structured form, paving the way for their eventual integration.

Given the validity and benefits of using a SW modelling approach, as shown in the previous chapter, this chapter proposes the *SOSENS (Social SENSor)* high-level ontology which provides a novel way to describe Social Sensing environments based upon state-of-the-art SW ontologies and conventions. SOSENS provides the basis for semantic social sensor & observation provenance, identification and linking, taking advantage of the multiple benefits of OWL & RDF.

Due to its ontological nature, SOSENS can easily be extended using ontologies that act as “plugins” and deal with other common aspects of Social Sensing environments, such as Data Quality. To this effect, (i) a Generic Value ontology and (ii) *SOSENS-Trust* is presented herein, with the latter acting as a high-level implementation-agnostic scaffold for describing Data Quality processes which are of great importance to a Social Sensing scenario.

### 3.1 Modelling Hardware Sensors

Before seeing how Social Sensors can be modelled using SW technologies, this section explores the nature and characteristics of Hardware Sensors and categorises them for the purposes of this thesis. It then explores available solutions to semantically modelling the above and proposes a solution that is fit-for-purpose.

#### 3.1.1 Nature and characteristics of Hardware Sensors

In the “classic” sense, a sensor (also called detector) is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or

by an (electronic) instrument. This can be as simple as a temperature sensor periodically recording the value of the outside environment, or as complex as a device that integrates multiple sensors in one package.

The BodyMedia SenseWear [71], for example, is a commercial multi-sensor designed for Energy Expenditure Monitoring. The sensor is capable of determining *when a user performs physical activities and how much energy is expended* during that period. To assess that, it uses multiple sensors (see **Figure 11**). The output, reading or measurement of the sensor (called a *sensor observation* for the remainder of this document to align with conventions used in Semantic Web ontologies) can be the output of each “sub-sensor” individually or the *overall* computed amount of energy (here: in kcal) expended during physical activity.



- *Motion Sensing* via an Accelerometer
- *Step Counter* via an Accelerometer
- *Galvanic Skin Response* (measures the electrical conductivity of the skin, which changes in response to sweat and emotional stimuli)
- *Skin Temperature* via a sensitive electronic thermometer
- *Heat Flux* (measures the amount of heat dissipating from the body)

**Figure 11:** The BodyMedia SenseWear Sensor & its sensing capabilities

Most sensors are designed to measure specific quantities only – they have specific Measuring Capabilities [70]. As such, some of the aspects of a specific sensing scenario may not be covered in their entirety by the available hardware sensors. In the Motivational Scenario of Chapter 1, for example, a SenseWear is incapable of sensing the *type* of physical activity performed.

In general, data gathered in a sensing environment may come from one or many, same or heterogeneous sensors, which means that sensor observations may come in varied formats. For example:

- A person can be measured by one or more different sensors which measure the same attribute in different way (e.g. one sensor can measure energy in kcal and another in joules or in an even higher-level concept such as steps per minute).
- A human sensor may express the same event as a hardware sensor, albeit in a textual manner (e.g. “moderately active”).
- The output of a sensor can be time-stamped in time instants or time intervals.

This creates the definitive need for data to be accompanied by proper semantic metadata in order to ensure usability and facilitate integration. For the purposes of this thesis, a categorisation of sensor data & metadata to be modelled in a sensing scenario, in the *broadest sense*, is as follows (see **Figure 12**):

- *Sensor Information* includes a sensor’s ID, Manufacturer, Model and other related concepts.
- A sensor’s *Measuring Capabilities* defines *what* is measured and *how*, including variables such as accuracy, sampling times and data acquisition methods.
- A sensor also has *Operating Restrictions* which denote under which conditions the sensor will produce expected results.
- In the case of Sensor Networks, the *Network Structure* refers to where and how sensors are deployed and connected.
- Finally, *Sensor Observations* (or Sensor Measurements) is the output of the sensor.

Available Sensor Data	
<ul style="list-style-type: none"> <li>▪ Sensor Information</li> <li>▪ Sensor Measuring Capabilities</li> <li>▪ Sensor Operating Restrictions</li> <li>▪ Sensor Network Structure</li> </ul>	<ul style="list-style-type: none"> <li>▪ Sensor Observations</li> </ul>

**Figure 12:** Sensor data & metadata to be modelled in a sensing scenario

### 3.1.2 Semantic Representation

As discussed in Chapter 2, ontologies are the main building block of the Semantic Web, thus an ontology is needed for representing the data presented in **Figure 12**. When it comes to fulfilling this goal in the SW, “re-inventing the wheel” by creating a new ontology is not only time-consuming but also discouraged. Thus, re-using existing ontologies where possible not only *promotes the SW vision* but also creates the necessary conditions for and facilitates *linked data discovery*.

#### Exploring available solutions

In related literature, early ontologies describing sensors and their properties were numerous but the research space was mostly characterised by the following issues [70], [72]:

- most ontologies were project-specific
- they did not see much re-use apart from the specific project
- no alignment between them
- no ontology best design practices

To tackle this problem, the W3C Semantic Sensor Network Incubator Group began work on SSN, the *Semantic Sensor Network* ontology, which was finalised and published in 2012 [70]. The group, comprised of 18 regular members as well as contributors from 19 organisations and 4 invited experts, reviewed existing ontologies and, taking into account data encodings and Web services to store and access sensor-related data such as the OGC’s Sensor Web Enablement (SWE) platform, designed a new, high-level sensor ontology. The semantic annotations of SSN are designed to improve interoperability and integration in sensing environments and, due to their abstract nature, can act as a base for developing related SW applications. Furthermore, the alignment between the SSN ontology and the DOLCE Ultra Lite [73] upper ontology normalised the structure of the ontology which allows its use in conjunction with ontologies or linked data resources developed elsewhere. Finally, other approaches such as the OGC SWE and SSN-XG are not exclusive of one another. The SWE is intended to provide standardization at the syntactic and service levels and does not address semantic level interoperability. For example, existing SWE technologies can be extended to support semantic metadata through annotation [74]

The SSN ontology’s use has been widespread since its introduction (see [75]) and is actively maintained & developed until the time of writing, with work being continued in the “Spatial data on the Web” Working Group [76]. Given all the above, it can be considered a perfect *candidate for re-use* in this thesis, to be used

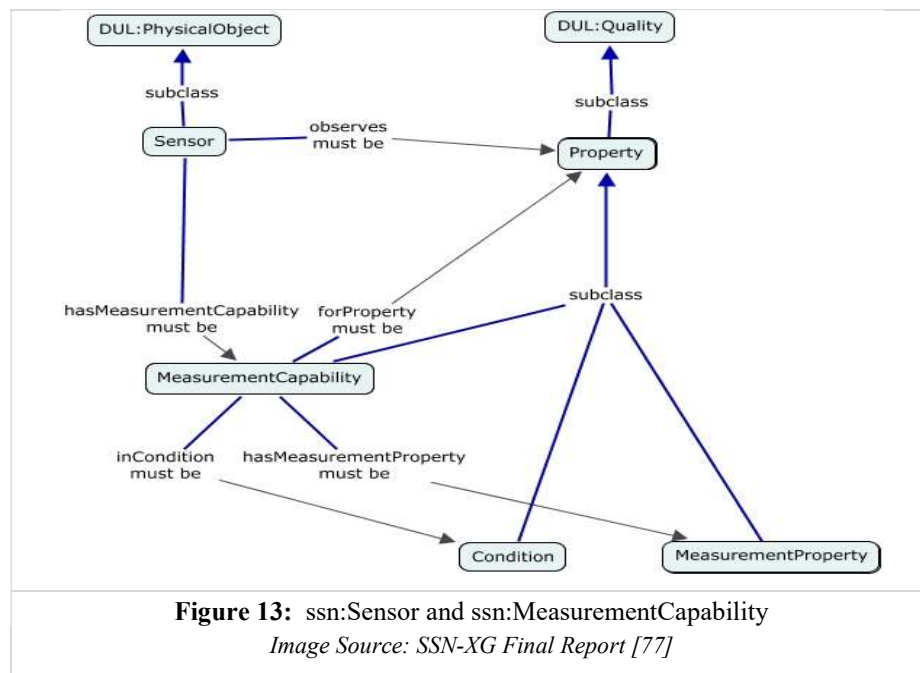
as the *basis* for describing Sensing environments. The chapter that follows briefly presents the basics of the SSN ontology.

### 3.1.3 The SSN (Semantic Sensor Network) ontology

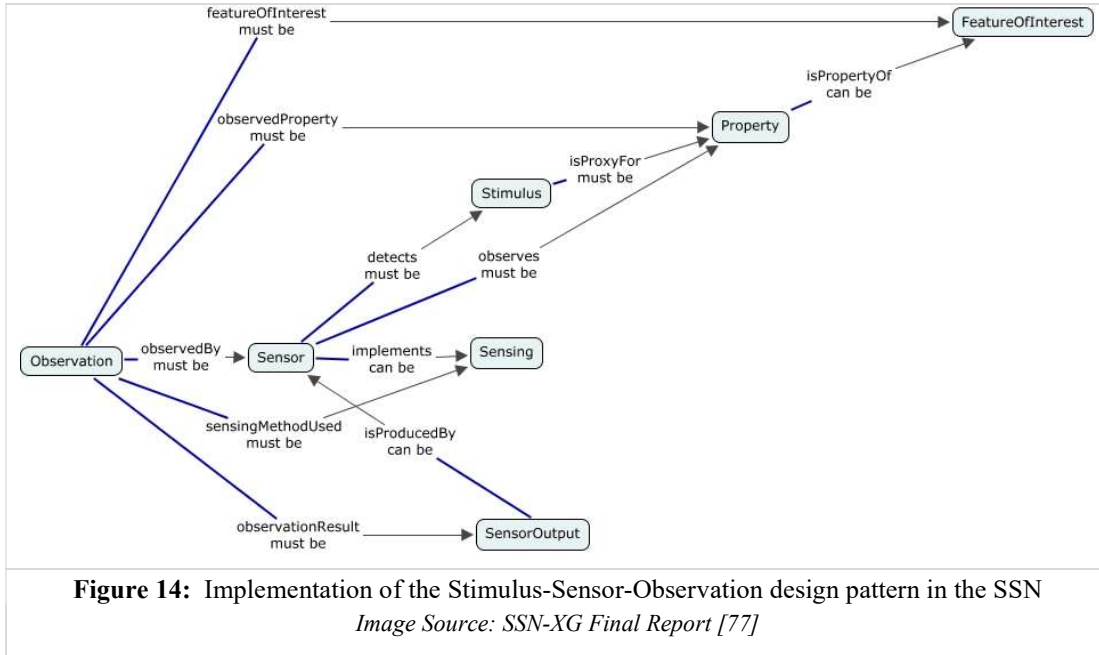
The SSN ontology is based around the concepts of (i) systems, (ii) processes, and (iii) observations [77]. One of its main features, which make it an exceptionally good fit for the work in this thesis, is that sensors *are not constrained to physical sensing devices*; rather, a sensor is *anything that can estimate or calculate the value of a phenomenon*. In other words, a device, a computational process, a human or a combination of the above could play the role of a sensor. The representation of a sensor in the ontology links together what it measures, the sensor and its functions and the processing.

It should also be noted that SSN is *intentionally* abstract in many “common” features of other ontologies, such as declaring time and observation values. This is a feature of the ontology rather than a limitation, as it provides the appropriate high-level approach needed to describe heterogeneous sensory environments.

First and foremost, the SSN ontology semantically declares sensors as things that can observe one or more *Properties* (i.e. a thermometer observes temperature). For these properties, each sensor has one or more *Measurement Capabilities* which describe how and under which conditions these properties can be measured, as seen in **Figure 13**.



Using the Stimulus-Sensor-Observation Design Pattern [70], SSN semantically annotates an *observation* made by a *sensor* for a *property* of a specific *feature of interest* (e.g. the area defined by geo-location X,Y), using the “observation” module as seen in **Figure 14**. It should be noted that observations in SSN simply act as the nexus between incoming stimuli, the sensor, and the output of the sensor, i.e. they are social, not physical objects. The actual value of an observation is described in the Sensor Output module.

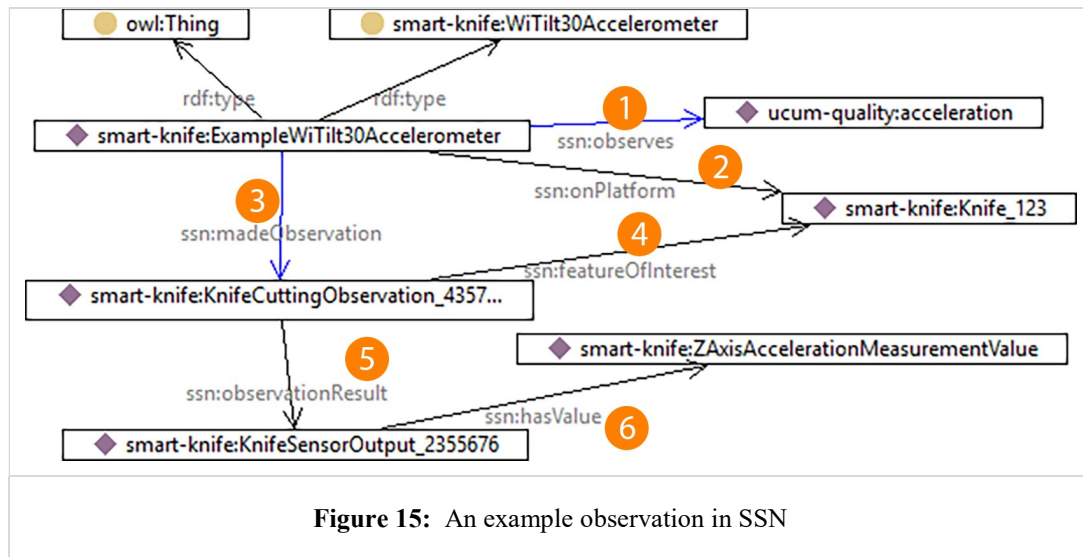


For example, consider a “smart-knife”, a member of the SmartProducts project [78] family which sports an accelerometer on its base. **Figure 15** shows how SSN can be used to express the observations of such a device<sup>5</sup>:

- 1 The sensor `ExampleWiTilt30Accelerometer` observes the property `ucum-quality:acceleration`. Note how SSN does not define any predetermined way to describe properties of objects and, thus, an external ontology is used (in this case, the Unified Code for Units of Measurement of the MUO ontology [79]).

<sup>5</sup> The same example, albeit in its original RDF form can be found in the SSN-XG documentation. The screenshot provided is from its import into the TopBraid Composer software and its further class/instance/property ontological visualisation.





- 2 The sensor is declared to be on the `Knife_123` platform (an example instance of the Knife smart product)
- 3 The sensor made an observation which is an instance of the `ssn:Observation` class
- 4 The observation is “about” (i.e. has an `ssn:featureOfInterest`) the `Knife_123`
- 5 The result of the observation is an instance of the `ssn:SensorOutput` class
- 6 The actual value of the observation is declared via the `ssn:hasValue` property. Note that, as was the case with the observed property, SSN does not define a way to express values, and a custom class is used.

As it can be seen, SSN uses an abstract but straightforward way to model sensors and their accompanying (meta)data in a SW environment. The information & example above obviously represent a small part of the ontology’s capabilities and is only included here as the basic subset of documentation needed for this thesis. For more information, the reader is referred to the main documentation of the ontology, which can be found online at:

<http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/>

### Proposed solution

Given that SSN provides a complete solution to the problem of semantically modelling hardware sensors, it can be used as-is for the purposes of this thesis. Not

only that, but it can serve as the basis for semantically describing Social Sensors as can be seen in the section that follows.

## 3.2 Modelling Social Sensors

This section explores the nature and characteristics of Social Sensors and categorises them for the purposes of this thesis. As discussed previously, a Social Sensor can be defined as a Human Sensor posting its observations on Social Media (SM).

However, the term “Social Sensor” is a philosophical, computer-science-engineered description of the above. In reality, no user sets out to be a Social Sensor specifically<sup>6</sup>; each user has his/her own volitions when posting something on Twitter or filling out a Facebook or PatientsLikeMe.com profile and definitely does not consider themselves a Sensor, even if they involuntarily become one.

Thus, this chapter begins by exploring the nature and characteristics of data on Social Media in general, “every-day” life and explores available solutions to semantically modelling data in a way that is abstract and facilitates their re-use. It then proposes a custom solution, inspired from the SSN ontological patterns, which semantically describes SM users as Social Sensors.

### 3.2.1 Nature and characteristics of data on Social Media

According to literature, Data on Social Media such as Social Networking Sites can generally be viewed as a *twofold structure* [80]:

- data describing the social network structure (Social Data) and
- data describing the content produced by network members

In order to encapsulate all the available properties of data on SM, this thesis makes a broader categorisation on the above by defining two categories:

1. **User-Generated content (UGC):** content produced by network members regularly such as Facebook wall posts, Tweets, microblog posts etc.
2. **Platform data:** the data that already exists in the platform, independent of the User’s UGC activity. This includes:

---

<sup>6</sup> Excluding, of course, incentivized social sensing experiments where users are specifically asked to report observations

- a. The user's **Social Structure** (friends, connections, groups, communities etc.) and
- b. The user's **Profile Data**.

A further categorisation can be performed for Profile Data according to their Dynamic & Static aspects:

- **Static Profile Data** includes profile data that are immutable such as Hometown & Gender
- **Dynamic Profile Data** includes data such as Age, Location and possible implementation-specific (e.g. in a Nutrition-specific SNS) profile data such as Height and Weight.
  - o i.e. data that could be considered of a self-reported observation-like nature (self-reporting location, self-reporting height)

**Figure 16** presents a collective view of the above with examples for each category.

Available Social Networking Site Data				
Platform Data			User-Generated Content	
Profile Data		Social Structure	<ul style="list-style-type: none"><li>▪ Facebook Wall Posts</li><li>▪ Events</li><li>▪ Pages</li><li>▪ Notes</li><li>▪ Memos</li><li>▪ Uploaded pictures</li><li>▪ Links</li></ul>	
Static Data	Dynamic Data			
<ul style="list-style-type: none"><li>▪ Name</li><li>▪ Hometown</li><li>▪ Gender</li></ul>	<ul style="list-style-type: none"><li>▪ Location</li><li>▪ Age</li><li>▪ Height</li><li>▪ Weight</li></ul>	<ul style="list-style-type: none"><li>▪ Social Connections</li><li>▪ Groups</li><li>▪ Communities</li><li>▪ Circles</li></ul>		
<b>Figure 16:</b> Examples of Social Sensor data & metadata to be modelled				

For example, suppose a user which begins to set up a Facebook profile. When signing up, he fills his profile with information describing him, such as where he lives, his gender, age and other information. Some of these, like age, are changed dynamically by Facebook if the user uses his birthdate as an input, so these form part of the “Dynamic Data” set. Also, it's common for people to move from location to location (e.g. to study abroad) and update their Facebook profile, which also causes a change in the “location” variable (also part of the Dynamic Data set). Other data, like Hometown, which statically define a user and are not expected to change, can be considered static data.

When a user starts using Facebook, he practically begins creating two things:

- The user's UGC, which is the wall posts, comments, events, pages notes and all other Facebook-defined constructs
- The user's Social Structure, which is the other users he decides to "friend" or "follow", the "groups" or "pages" he will join etc.

### 3.2.2 Semantic Representation

As was the case with the semantic representation of hardware sensor, this section first explores available ontological solutions for describing the data presented in the previous section.

#### Modelling Static Profile data: FOAF

FOAF (an acronym of **F**riend-**O**f-**A**-**F**riend) is an OWL ontology describing persons, their activities and their relations to other people and objects. FOAF is broadly considered as the first Social Semantic Web application, and is frequently used in books & tutorials to showcase the versatility of OWL & RDF.

Coming back to the previous section's categorisation of SM data, FOAF can be used in its most basic form to describe users and their static profile data. With properties such as `foaf:name`, `foaf:mbox` & `foaf:mbox_sha1sum` (for privacy reasons), FOAF is the de-facto choice for describing users on the SW. For example, the RDF below describes the author:

```
:me
a foaf:Person ;
foaf:name "Ioannis Pagkalos" ;
foaf:firstName "Ioannis";
foaf:lastName "Pagkalos";
foaf:nick "ipagkalos" ;
foaf:mbox_sha1sum "b34535f6a830086d662a8[...]c970" ;
foaf:homepage <http://ion.pagkalos.com> ;
foaf:workplaceHomepage <http://www.ee.auth.gr> ;
```

RDF/Turtle

## Modelling Social Structure: FOAF

FOAF is also one of the key players in “decentralising” data describing users’ Social Network [40], away from “silo” websites<sup>7</sup> such as Facebook. The ontology specifies relationships between users in a basic, machine-understandable way. FOAF profiles can be generated and gathered from other services in order to integrate users’ social data from various sources. Many people already keep a personal FOAF profile on their website or on a shared hosting space and there are also SNSs that export FOAF [81].

Coming back to the social structure modelling problem, let  $U = \{u_1, u_2, \dots, u_n\}$  be a group of  $n$  SM users. In Social Network theory, a graph  $G = (V, E)$  where  $V$  a finite set of vertices and  $E$  a finite set of edges such that  $E \subseteq V \times V$  can be used to model the social network that describes  $U$  [68]. Furthermore, the matrix  $M$

$:= (m_{i,j})_{n \times n}$  where  $m_{i,j} = \begin{cases} 1 & (u_i, u_j) \in E \\ 0 & \text{otherwise} \end{cases}$  and  $n = |V|$ , associated with that graph, is used to describe the relationships between individuals (reciprocal or not).

Using FOAF, a tie ( $m_{i,j} = 1$ ) is modelled as a `foaf:knows` relationship between  $u_i$  and  $u_j$ , which are both instances of `foaf:person`. The RDF below shows an example of these simple assertions:

`:me`

RDF/Turtle

```
a foaf:Person ;
foaf:firstName "Ioannis"^^xsd:string ;
foaf:lastName "Pagkalos"^^xsd:string ;
foaf:knows :Dave_Jones ;
foaf:knows :Jane_Smith .
```

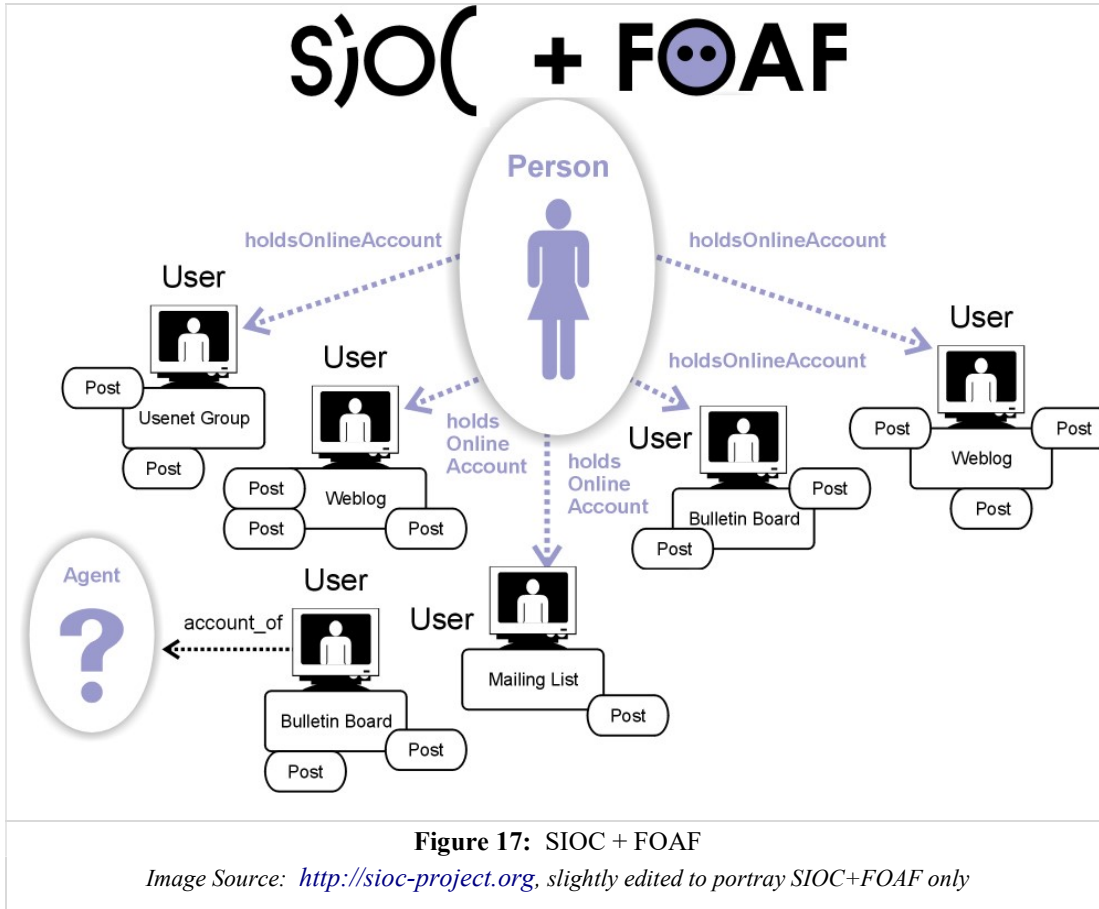
Following the same principles, a social group  $SG_i$  (a community, group of interest etc.) that consists of users  $u_i \in U$  is modelled as a `foaf:group` that has as `foaf:member` one or more `foaf:person`. This provides an abstract approach towards a user’s social data that can be further refined if needed. Extensions to FOAF that more clearly define relationships and tie strengths can be custom-built or found as published ontologies (e.g. [82],[83],[84]) which further proves the power of FOAF as a starting point for modelling social data.

---

<sup>7</sup> “Silo” websites: sites which form “information silos”: Information on one site is not usable in the other

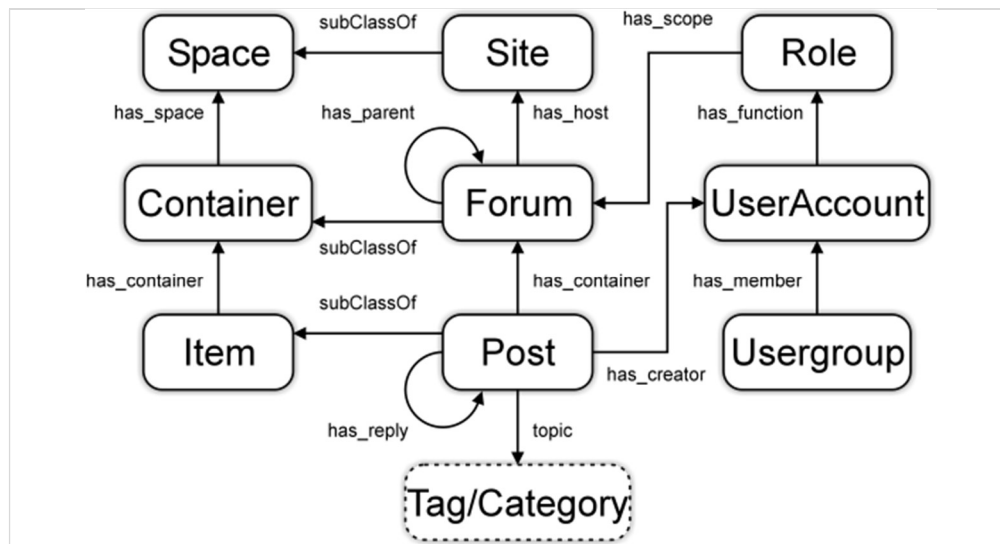
## Modelling UGC: SIOC

In order to model the User Generated Content (UGC), the SIOC ontology (Semantically-Interlinked Online Communities) is another staple of the Social Semantic Web. It is already designed with FOAF in mind (see **Figure 17**) and semantically defines users which create content through their “online personas”, their `sioc:UserAccount`.



It should be noted that SIOC is not designed for Social Media specifically, as it also encompasses other forms of online communities such as fora and mailing lists. This, however, does not discourage its use on Social Media, as the core SIOC concepts, as seen in **Figure 18**, can be easily applied there.

For example, in SIOC concepts:



**Figure 18:** The SIOC core ontology

*Image Source: SIOC Core RDF spec [85]*

- Facebook is a `sioc:Site` which is the space (`sioc:has_space`) for
- `sioc:Container(s)` such as Walls, Groups, Pages etc., where
- `sioc:Post(s)` can be created by someone (a `foaf:person`) who owns a
- `sioc:UserAccount` on Facebook

If, for example, the author was to create a post on his Facebook wall, then the RDF using FOAF/SIOC would look something like this:

```

:Facebook a sioc:Site ;
  rdfs:comment "The SNS \"Facebook\" at www.facebook.com".

:ipagkalos_Facebook a sioc:UserAccount
  sioc:account_of <http://ion.pagkalos.com/#me> .

:ipagkalos_FacebookWall a sioc:Container ;
  sioc:has_space :Facebook ;
  rdfs:comment "The Facebook wall of ipagkalos" .

:ipagkalos_wallPost_1451649600 a sioc:Post ;
  sioc:has_container :ipagkalos_FacebookWall ;
  sioc:has_creator :ipagkalos_Facebook ;
  dct:created "2016-01-01T12:00:00" ;
  sioc:content "It is so sunny outside!" .
  
```

## From UGC to observations: The Social Sensor paradigm

As discussed in the introductory chapters, the main position of this thesis is that many times, users' generated content on Social Media (and, in some cases, parts of their Dynamic Profile Data) are about sensed events, and as such, can be treated as observations by human sensors. In such cases, human sensors can be considered **Social Sensors**, as they operate in a social environment (a "Social Sensing" environment), as part of a Social medium. This adds specific properties to them, which need to be appropriately modelled in machine-understandable SW terms. In addition, human sensors who act as social sensors need to be brought to the same semantic level as a hardware sensor while at the same time making sure that important social media metadata is not "lost in translation".

Seeing that (a) FOAF & SIOC create a comprehensive basis for describing users & their UGC in Social Sensing environments and that (b) SSN is a solid basis for describing all sensors, a "semantic combination" of the above along with appropriate ontological patterns and guidelines forms a concrete solution for modelling Social Sensing environments using SW technologies. The chapter that follows presents the high-level ontology that was designed exactly for that purpose.

## 3.3 The Social Sensor (SOSENS) Ontology

The proposed SOSENS (**SO**cial **SENS**or) Ontology is an OWL2-DL ontology which provides a novel way to describe Social Sensing environments based upon state-of-the-art SW ontologies and conventions. SOSENS provides the basis for semantic social sensor & observation provenance, identification and linking, taking advantage of the multiple benefits of OWL & RDF, while maintaining an abstract, high-level modelling approach to the problem.

SOSENS builds upon the Social Sensor paradigm, where human sensors share their observations on Social Media. Instead of focusing on users and their UGC, SOSENS uses the sensor & observation semantics as the core ontological concept, by using SSN as the starting design point of the ontology.

### Putting it all Together

SOSENS re-uses (imports) the three well-established ontologies presented before, and links them together, providing the "*semantic scaffold*" where needed for their interoperability in a Social Sensing environment:

1. SSN (Semantic Sensor Network ontology) is used to describe the Social Sensors, what they measure and the result of these observations.



2. FOAF (Friend-of-a-Friend ontology) is used to describe the human behind an observation as well as his/her social network
3. SIOC (Semantically Interlinked Online Communities ontology) is used to describe the UGC of the user which act as the source statements for the observation

However, some “*semantic glue*” is still needed. Apart from linking and re-using these ontologies in a way that is meaningful for Social Sensing spaces, another important part of SOSENS is handling the *correlation* between a `sloc:Post` and its semantic expression as a `ssn:Observation`. It is important to note that this refers to how the information from a `sloc:Post` was converted to an `ssn:Observation` and *not* how the actual estimation of the event was computed by the sensor<sup>8</sup>; in other words, focusing on who, how and when it was decided that a specific post on Facebook is actually an observation by a human sensor.

To aid in this respect, SOSENS introduces the concept of the `sosens:ParsingActivity` in order to keep track of how a `ssn:Observation` came to be, along with respective classes & properties that describe this procedure. Keeping in line with the “re-use ontologies where possible” philosophy of the SW, SOSENS imports a final core ontology to describe provenance:

4. PROV-O (Provenance Ontology) is used to describe parts of the provenance chain of a Social Sensor observation

The PROV Family of Documents [86] defines a model, corresponding serializations and other supporting definitions to enable the inter-operable interchange of provenance information in heterogeneous environments such as the Web. PROV-O (the **PRO**Venance **Ontology**) defines a light-weight OWL2 ontology for the provenance data model, which provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts. Quoting the authors, “[*PROV-O*] can also be specialized to create new classes and properties to model provenance information for different applications and domains”; SOSENS does exactly this to model Social Sensing environment provenance:

In SOSENS, a `sosens:Parser` (a `prov:Agent`) is associated with activities of the type `sosens:ParsingActivity` (a `prov:Activity`) that “convert” a

---

<sup>8</sup> This relates to a sensor’s measurement capabilities

`sioc:Post` to a `ssn:Observation` (both a `prov:Entity`), using a specific `sosens:ParsingMethod`. In reality, there are many different ways to do this parsing<sup>9</sup>, which is why SOSENS does not include a one-size-fits-all solution, but rather focuses on making sure that, whichever process is followed, it is documented via these OWL constructs. An example of such a process is provided in the pilot implementation of this methodology, presented in **Chapter 5**.

Keeping track of the `sosens:ParsingActivity` is critical for multiple reasons. For one, parsers can often misidentify or misinterpret the UGC of a user so it is important to be able to backtrack to the actual original statement. If this is a continued behaviour, this can lead to observations that were parsed by a specific `sosens:Parser` being excluded by data consumers. Additionally, according to application needs and the type of the sensed event, one source statement may refer to multiple `ssn:Observations`. For example, a post stating that “the weather is nice outside” or “I can see a fire out my window” can be treated as an observation of the Social Sensor for that particular moment. On the other hand, a reported hour of jogging could be broken down to multiple observations, each representing a different moment in the time period. In such cases, a `sosens:ParsingActivity` is a helpful link back to the source statement.

Finally, apart from the basic SIOC classes, SOSENS introduces some minor subclasses of SIOC such as `sosens:SocialNetworkingSite` and `sosens:SNSData` to better categorise data. However, the primary purpose of SOSENS is to act as a “semantic scaffold” for Social Sensing environments, following the “less is more” approach when it comes to creating new properties & classes. Therefore, the number of these created sub-classes & properties is limited.

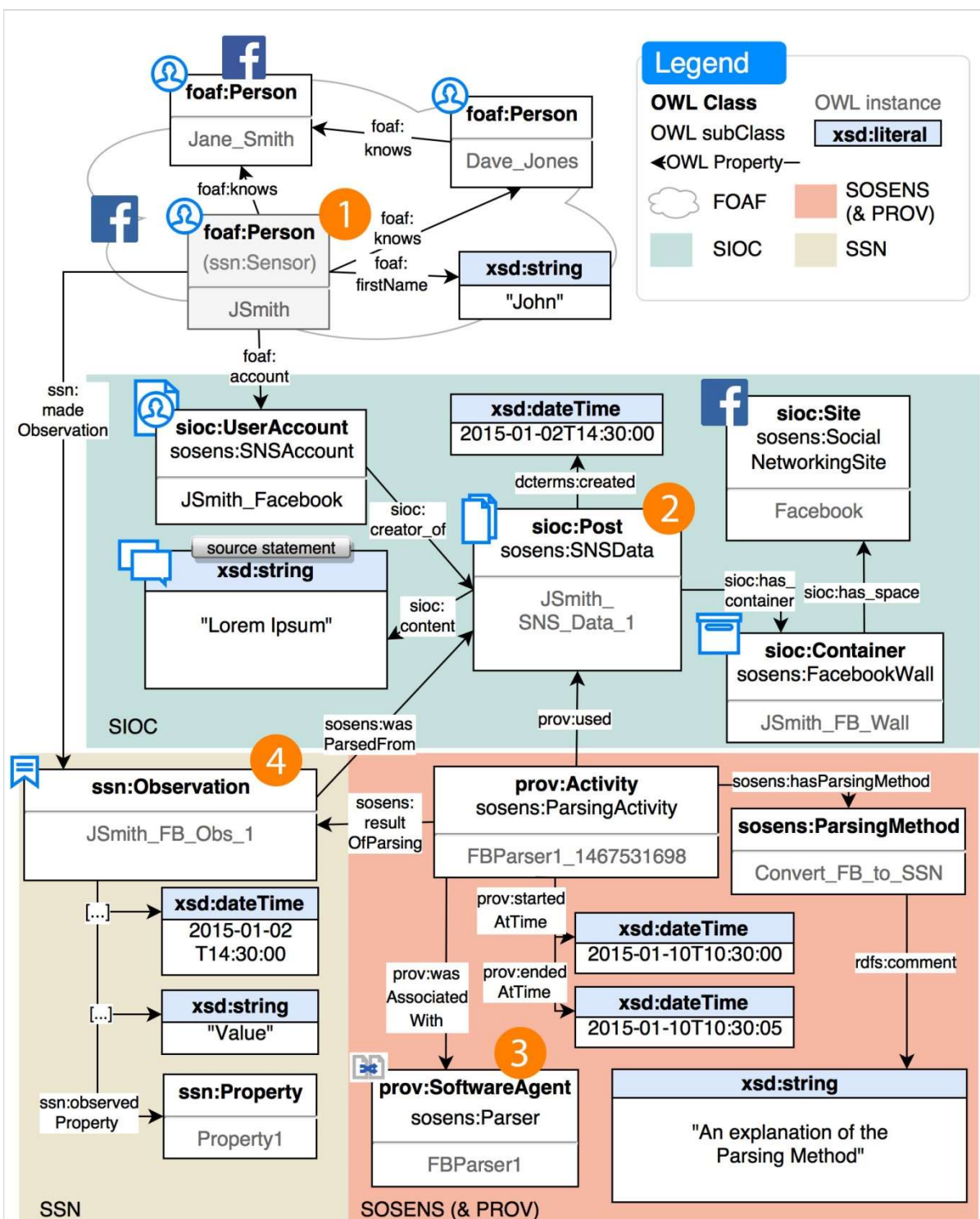
Because ontologies are better described through visualisation, **Figure 19** shows how SOSENS can be used to describe a Social Sensing scenario in OWL constructs<sup>10</sup> (classes, subclasses, properties and instances).

*In this particular example, “John Smith” is posting his observations on Facebook about a specific monitored property - “Property1” (e.g. his physical activity). Using SOSENS, this is expressed in SW terms as follows:*

---

<sup>9</sup> In addition to the multitude of available methods on event detection (e.g. [15], [87]), emotion detection (e.g. [88]) etc.

<sup>10</sup> The observant reader will notice that this is a more detailed description of **Figure 1**, using SW terms



**Figure 19:** Describing a Social Sensor observation on Facebook using the Social Sensor (SOSENS) Ontology

- 1 JSmith is a `foaf:Person` & `ssn:Sensor`, whose Social network on Facebook (e.g. his friendship status with Jane Smith) is described via `foaf:knows` properties.
- 2 Jsmith, through his Facebook Persona `#JSmith_Facebook`, creates a `sioc:Post` on a `sioc:Container` (here: his Facebook Wall) with some content, which is treated as the *Source Statement*. The post is accompanied by metadata such as time of creation, and, of course, its content in the form of an `xsd:string`.
- 3 A `sosens:Parser`, here semantically defined as `FBParser1`, “picks up” the post and converts it to an `ssn:Observation` using a specific `sosens:ParsingMethod` (here: `Convert_FB_to_SSN`). This method is accompanied by metadata such as the `prov:startedAtTime` and `prov:endedAtTime` and provides incoming (source) and outgoing (output) semantic links to the `sioc:Post` and `ssn:Observation` respectively. For the former, the `prov:used` property is used for the link, while for the latter, the `sosens:resultOfParsing` property is used.
- 4 In regards to the actual observation itself, Jsmith, is defined as the source (`ssn:madeObservation`) of an `ssn:Observation` which includes the typical SSN properties such as `ssn:hasValue` and `ssn:ObservationResultTime`, based on the original `sioc:Post`<sup>11</sup>. This “semantically” (i.e. via a reasoner) makes Jsmith a `ssn:Sensor`.

SOSENS’s approach to Social Sensors creates an adaptable ontology, easily re-useable in many Social Sensing scenarios. As is the case with most of the high-level ontologies (and the purpose of a ‘semantic scaffold’), SOSENS intentionally leaves some things to be decided on the application level. For example, the details of how a `sosens:ParsingMethod` actually operates are represented in the figure as a simple `rdfs:comment`. In other scenarios, these can be explained in a much more verbose way, by adding a new class and properties, or even linking to other ontologies that describe such methodologies.

---

<sup>11</sup> The details of how an observation is expressed in SSN are not shown in the figure due to space constraints, but more information can be found on the SSN ontology description [70] or an example can be seen in **Chapter 5**.

## 3.4 SOSENS Extensions

One of the major benefits of using a Semantic Web approach to model Social Sensing environments is the extensibility provided by OWL and RDF. Seeing how easily FOAF, one of the most basic publically-available ontologies, has been extended by third-parties (see **Chapter 3.2.2**) showcases the low degree of effort required to “upgrade” and modify an ontology with added features. A common practice for extending an ontology is to find a broadly-defined term in it and start from there, by defining more specialised and/or application-/domain-specific classes & properties. For example, extending `foaf:knows` with several more “degrees” of friendship, or family relations is as simple as creating an ontology which imports FOAF and adds these classes.

The same principles apply to SOSENS, which is intentionally designed with quite a few abstract terms that are candidates for extending. Some, like the lack of an explicit way of expressing observation values, come from the core ontologies on which SOSENS is based, while others, like building upon measurement capabilities and sensor accuracy to define sensor trust & reputation, introduce new concepts and their respective ontological modelling.

The chapters that follow showcase two extensions for SOSENS which accommodate the two issues presented above.

### 3.4.1 SOSENS Generic SSN Value

The first extension to SOSENS is a very simple, generic family of `ssn:observationValue` classes in order to get a user started with (a) expressing (Social) Sensor data in a Semantic Web environment and (b) extending the SOSENS ontology. It describes observations with *quantity* or *quality* values and is based on (a) the smart-knife example, as presented in the SSN-XG Documentation and (b) the 2009 position paper by Sabou et al [89].

The classes included are defined as follows (both are subclasses of `ssn:Observation`):

- **`sosens:QuantityObservationValue`**, with the following OWL *restrictions*:
  - `sosens:hasQuantityUnitOfMeasurement` exactly 1
  - `sosens:hasQuantityValue` exactly 1
- **`sosens:QualityObservationValue`**, with the following OWL *restrictions*:
  - `sosens:hasQualityValue` exactly 1

An example using the above, for a quantitative temperature observation value of 50.0° Celsius and a qualitative observation of “Warm” can be seen below:

```
:temp_Observation_1                                RDF/Turtle
  ssn:sensorOutput [
    ssn:hasValue [
      sosens:hasQuantityUnitOfMeasurement :Celsius ;
      sosens:hasQuantityValue "50.0"^^xsd:float .
    ]
  ]
```

```
:temp_Observation_2                                RDF/Turtle
  ssn:sensorOutput [
    ssn:hasValue [
      sosens:hasQuantityValue "Warm"^^xsd:string .
    ]
  ]
```

*Note: this extension is packaged along with the main SOSENS ontology at <http://phd.pagalos.com/sw/sosens>*

### 3.4.2 SOSENS-Trust

As discussed in the introductory chapters, Trust & Data Quality is of great importance to a Social Sensing environment. Trust has another important role in the Semantic Web, as agents and automated reasoners need to make trust judgments when alternative sources of information are available [90]. However, while hardware sensors can undergo calibration & testing and usually come with specifications that accurately reflect what they can measure and under which – typically strict – conditions (i.e. “a pre-determined trust rating”, human sensors do not follow the same rules [3], [91], as:

- Human reliability cannot be calibrated
- Humans are not usually trained for specific observations
- Humans may be emotional/have their own agenda when sensing
  - o Especially in Social Media
- Human sensors are not specialized; i.e. humans may have a multitude of sensing capabilities and thus any sort of reliability must be functional, for a specific context

Taking the above into account, it is, unfortunately, virtually impossible to design trust & data quality assessment methodologies for Social Sensors that would apply to all domains and Social sensing environments. Furthermore, trust is a highly subjective matter; a well-cited review paper on major areas of trust research [90] found 22 sub-categories of trust research (see **Figure 20**), spread across 101 research papers, each with their respective algorithms and trust methodologies.

<b>A categorization of major areas of trust research</b>	
<b>Policy-based trust</b>	<b>Reputation-based trust</b>
Network security credentials (1) Trust negotiation (9) Security policies and trust languages (11) Distributed trust management (4) Effect of credential type (1)	Decentralization and referral trust (9) Trust metrics in a web of trust (10) Trust in P2P networks and grids (5) Application-specific reputation (3)
<b>General models of trust</b>	<b>Trust in information resources</b>
General characteristics of trust (5) Computational and online trust models (6) Game theory and agents (5) Software engineering (1)	Trust concerns in the Web (2) Trust concerns in the Semantic Web (3) Trust using hyperlinks (4) Filtering information based on trust (3) Filtering the Semantic Web (4) Subjectivity analysis (3) Provenance information (5) Content trust (4) Site design and human factors (3)

**Figure 20:** A categorisation of major areas of trust research [90]. Numbers in parentheses indicate the amount of research papers associated with each sub-category

Given this abundance and heterogeneity of methods, this thesis attempts to approach the Trust issue using the following requirements:

- Independently of which Trust / Data Quality assessment methodology is used in a Social Sensing Environment, an extension to SOSENS must provide the appropriate provenance details and links to the SOSENS ontological constructs (i.e. Person, Sensor, Observation etc) describing the methodology and how it was applied.
- This extension must cater (i) to a sensor's independent, per-observation Quality as well as (ii) their overall Trust score as a sensor that observes a specific property.

Although trust ontologies in OWL do exist, there are no works presently that provide a “link back” to SSN, the appropriate provenance chains and/or focus on Social Sensing environments. The section that follows presents an extension to SOSENS which implements the requirement presented above while maintaining a high-level approach that can accommodate existing solutions.

## The SOSENS-T Ontology

Having Social Sensor observations described in SOSENS creates a solid basis for establishing sensor provenance and identification. When all user statements are `ssn:Observations` and, thus, have a common data format, it becomes easier to apply and document trust assessment methodologies such as comparing a Social Sensor’s output to that of a trusted source (usually a hardware sensor observing the same property). To describe this process, a complementary ontology to SOSENS was developed, entitled the *SOSENS-Trust Ontology* (SOSENS-T).

### *Modelling Social Sensor individual Observation Quality*

SOSENS-T is a high-level ontological scaffold that describes the process of a `sosens-t:TrustAuthority` (a `prov:Agent`) computing and assigning a `sosens-t:Rating` for an individual `ssn:Observation` by performing a `sosens-t:RatingActivity` (a `prov:Activity`). This activity has a start and end time (`prov:startedAtTime` and `prov:endedAtTime`) and uses (`prov:used`) data from the Social Sensing environment such as two or more `ssn:Observations` to generate (`prov:wasGeneratedBy`) a `sosens-t:Rating`. Each activity follows a `sosens-t:RatingMethod` in order to compute a `sosens-t:RatingValue`, linked to the `sosens-t:Rating`. Similar to `sosens:ParsingMethod`, describing the method and the possible values of the rating is left to be decided on the application level, as it is – especially in the case of observation quality – a highly subjective matter.

Finally, this rating is linked back to the observation by *re-using an existing SSN property*, `ssn:qualityOfObservation`. **Figure 21** shows an example where an `ssn:Observation` created by `Jsmith` is compared via `RatingMethod1` to an observation by another `ssn:Sensor`, `Sensor_1`, which is treated as ground truth. The resulting `Rating` describes the quality of this observation, which has a `RatingValue` of 0.7.

Using the SOSENS-T classes and re-using the PROV-O and SSN properties where possible provides a provenance-aware scaffold to model observation quality.



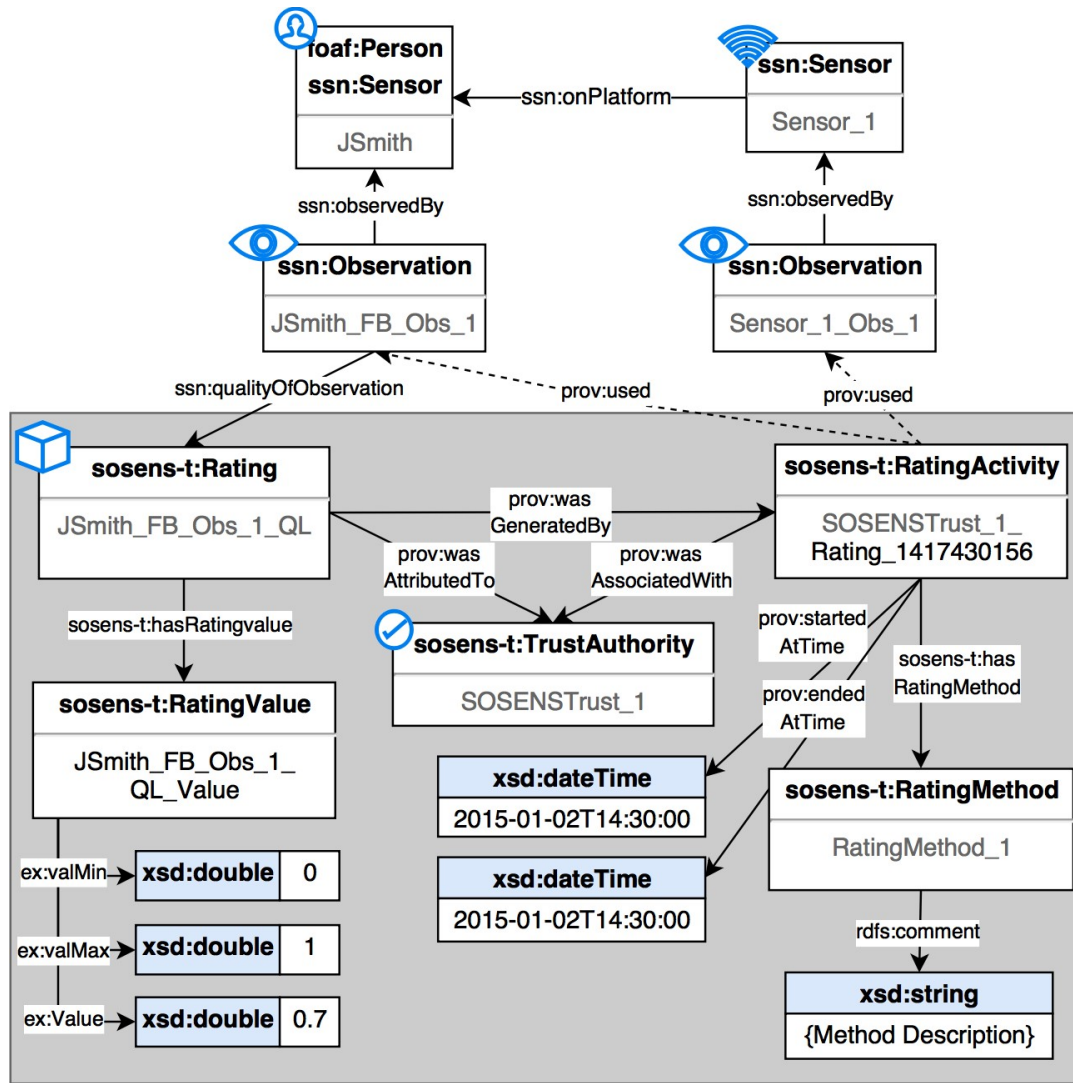


Figure 21: Describing observation quality in SOSENS-T

Instead of an arbitrary value of quality, through SOSENS-T a data consumer can easily determine:

- **how** and **when** this value was computed, via the `sosens-t:RatingActivity` and `RatingMethod`
- **who** was responsible for this quality assessment, via `sosens-t:TrustAuthority`
- the **result** of this assessment via the `sosens-t:RatingValue`

Even though these issues have been discussed extensively in many Computer Science fields, expressing them in a SW environment and linking them with PROV & SSN constructs creates a semantically-rich, machine-understandable environment which, given the rapid growth of the SW, is an important addition to

Social Sensing tasks. Because of RDF and OWL, the output of applications based on this methodology, like the pilot implementation presented in **Chapter 5**, can easily be shared and re-used. For example, implementations that use the same observation quality rating methods can be identified, semantically linked (via `owl:sameas` or OWL inferred classes) and its results aggregated or compared.

This approach also allows *existing* ontologies that describe data quality assessment methodologies, to be re-used, such as the DQM [92] ontology, whose data requirement classes can be treated as `RatingMethods` and its results as `:Ratings`.

### ***Modelling Social Sensor Trust***

Trust in a sensory environment is an equally delicate and subjective problem as that of determining individual observation quality. In *this work's* view of trust, the goal is to attach a *Trust Rating* to each Social Sensor so data consumers have an idea of how to treat the sensor's future observations. In a way, it is aligned with the view in [93]: “*In the context of the Web, trust translates to the belief that an information producer will create useful information*”.

Nevertheless, there is one dimension of sensors that remains constantly useful: *Accuracy*. For social sensors, accuracy can be treated as a measure of the degree that the data provided represents the correct state of the object or property measured. This definition agrees with the notion of Semantic Accuracy in [94] which is defined as “*the closeness of the value  $v$  to the true value  $v'$* ” as well as the description of `ssn:Accuracy` in the SSN ontology, which is “*The closeness of agreement between the value of an observation and the true value of the observed quality*”. Thus, a trusted (== accurate) sensor in the motivating scenario (see **Chapter 1**) would be one that has a high chance of actually being physically active when he/she says so on Facebook.

Because of its abstract nature, the SOSENS-T extensions can be re-used here to express such values in SW terms. Using its terminology, a Social Sensor's trust rating can be characterised by the `sosens-t:Rating` of its observations over time. Naturally, as was the case with individual Observation Quality, there are many computational ways to aggregate these ratings into an accuracy score (a thorough list is provided in [95]) so expressing them in SOSENS-T must be as abstract as possible. **Figure 22** shows how, in a similar way to how observation quality was modelled, a `sosens-t:RatingActivity` uses two `sosens-t:Ratings` from Jsmith's previous `ssn:Observations` in order to compute a new `sosens-t:Rating`.

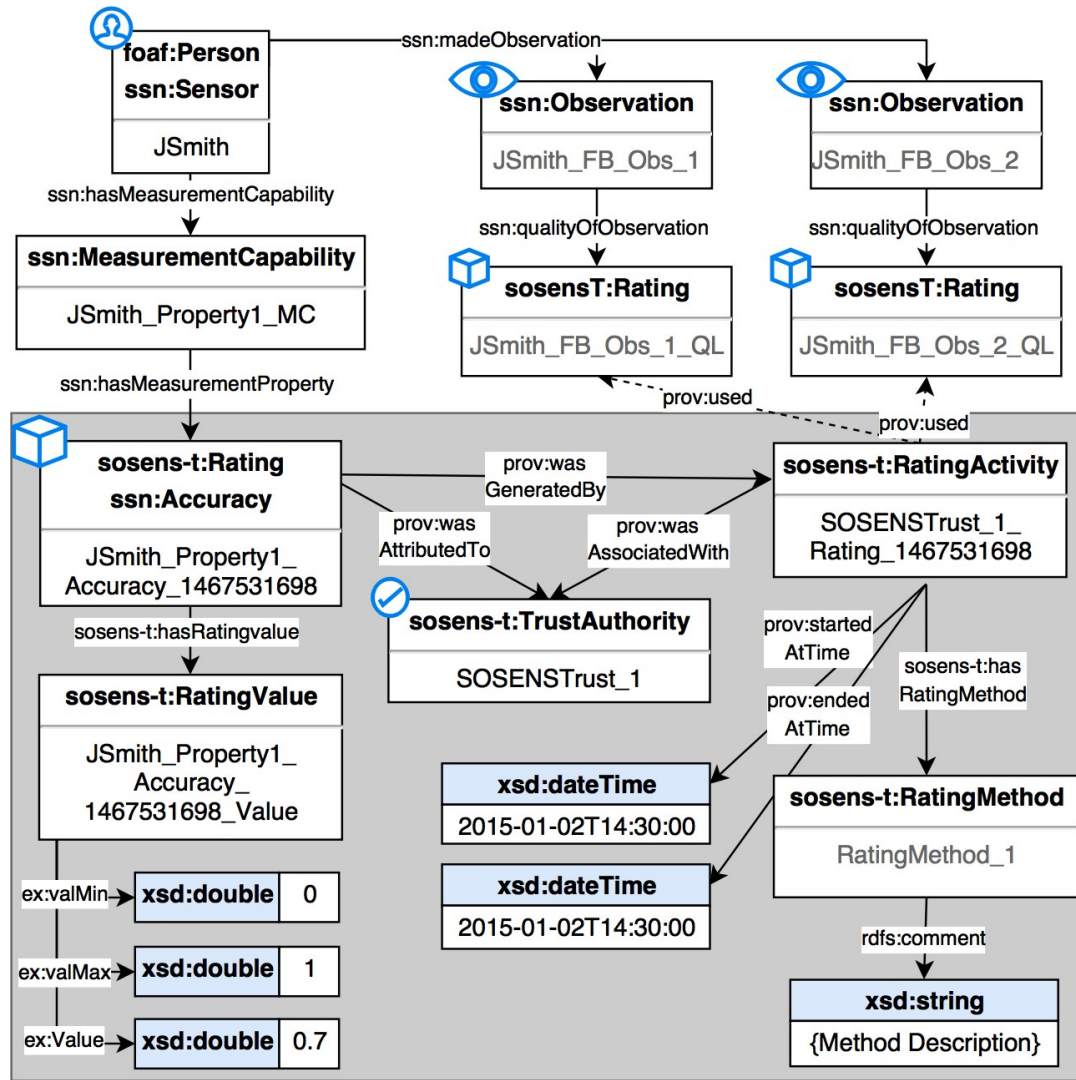


Figure 22: Describing Sensor Trust in SOSENS-T

An important aspect of this modelling is expressing, in SW terms, that this rating reflects the Social Sensor's accuracy in observing a specific property. SSN already defines an ontological structure where a `ssn:Sensor` has a `ssn:MeasurementCapability` for a specific observed property, which is then further characterised by a `ssn:MeasurementProperty` such as `ssn:Accuracy`. For example, "Jsmith can measure Property1 with a `ssn:MeasurementCapability` that has a `ssn:Accuracy` of X". In alignment with this structure, the only thing needed is for the `sososens-t:Rating` produced to be the "target" of the relevant `ssn:hasMeasurementCapability` property. Similarly to before, this rating has a specific rating methodology, is attributed to a Trust Authority and its actual value is decided on the application level.

Using this approach, even though Social Sensor trust is a highly subjective matter, SOSENS-T tackles the issue in the same way as with individual observation quality: making sure that provenance information for trust and its related concepts are expressed in SW terms, independently of the computational method used, in order to aid in decision-making and data re-use, both by machines and humans.

*Note: this extension is packaged separately than the SOSENS ontology and can be found at <http://phd.pagkalos.com/sw/sosens-t>*

## Chapter Conclusions

This chapter discussed the modelling aspect of Social Sensing environments using Semantic Web technologies by demonstrating how the combination of well-known ontologies using appropriate methods and novel features can result in a user in Social Media being described as a Social Sensor, while maintaining the same level of semantic detail & metadata as a hardware sensor.

To this effect, the *SOSENS* high-level OWL2-DL ontology was proposed. SOSENS provides a novel way to describe Social Sensing environments based upon state-of-the-art SW ontologies and conventions and can serve as the basis for semantic social sensor & observation provenance, identification and linking, taking advantage of the multiple benefits of OWL & RDF. Furthermore, SOSENS can be easily extended, as showcased by the two extensions presented within, with one dealing with the very important issue of Trust management in a Social Sensing environment.

# 4 The SOSENS Framework

## Chapter Introduction

The previous chapter presented the modelling of Social Sensing spaces using Semantic Web technologies and the SOSENS family of ontologies. The proposed approach combines (social) sensor identification, metadata & provenance under one semantic “umbrella” which makes it easier to express a Social Sensing environment in semantically-rich RDF as well as take advantage of all the benefits of a Semantic Web environment, such as machine-to-machine interaction and OWL inference.

This chapter presents the **SOSENS Framework**: a holistic, systemic and implementation-agnostic *software architecture* for modelling and managing Social Sensing environments based on the ontologies presented in the previous chapter. Using the framework and conforming to its requirements acts as a semantic “scaffold” for heterogeneous sensing environments and allows for the creation of novel, *interoperable* sensing spaces, as well as information that is reusable.

The framework is organised and presented loosely based on the Open Group Architecture Forum’s (TOGAF<sup>12</sup>) **Building Block** approach, which separates design from implementation by defining Architecture and Solution Building Blocks (**ABBs** , **SBBs**) respectively. Following this approach:

- *ABBs*: Capture architecture requirements & guide the development of SBBs
- *SBBs*: Define what products and components will implement the functionality while being product- and vendor-aware.

During the course of this chapter, *four* SOSENS ABBs as well as sample SBBs are introduced. The SBBs include a generic Web API that converts sensory data expressed in the common, easy-to-construct JSON<sup>13</sup> format to their respective SOSENS representation. The goal of the API is to make creating data in a SOSENS environment easy, even for data producers not overly familiar with SW technologies.

Finally, the contribution of the SOSENS framework in raising overall data quality is evaluated from the perspective of data consumers, using a list of 15 expected

---

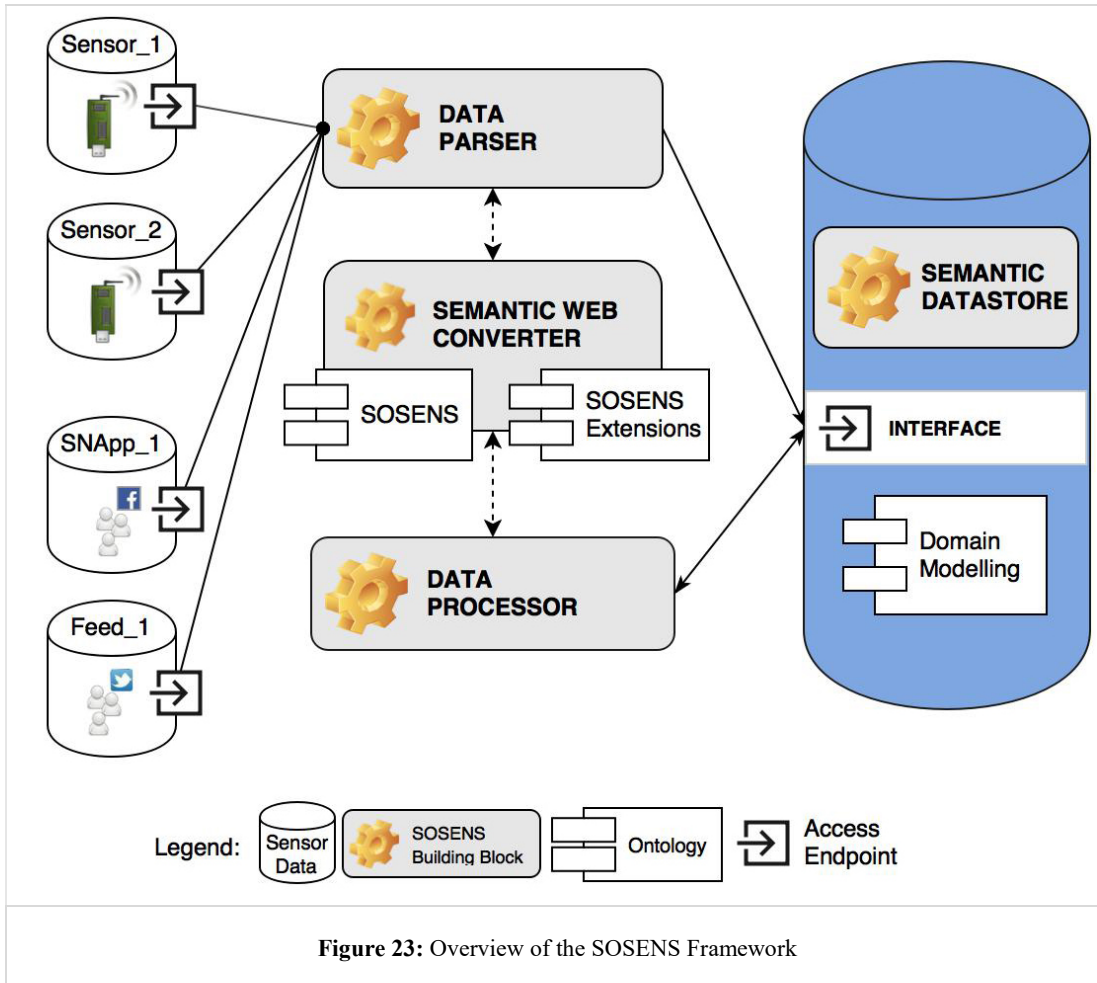
<sup>12</sup> <https://www.opengroup.org/togaf/>  
<http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html>

<sup>13</sup> RFC 4627 (JSON) - <https://tools.ietf.org/html/rfc4627>

data quality dimensions, and the performance of a reference implementation is tested in a commercial computer.

## 4.1 Framework Overview

The SOSENS framework is an implementation-agnostic Semantic Web architecture, which is based on the SOSENS ontology and its extensions.



**Figure 23** presents a “bird’s-eye” view of the framework as well as an example workflow:

In every Social Sensing environment, a variable amount of sensors are present, which produce and store data. As discussed before, these sensors may be Hardware Sensors (e.g. a Temperature Sensor, a Digital pedometer) or Social Sensors (e.g. an App on Facebook, a Twitter Feed). Architecturally, SOSENS is agnostic to the data communication procedures between sensor and data consumer (e.g. a sensor base

node) and assumes that each sensor records its observations in a temporary or permanent data store that is accessible via an endpoint. These endpoints can be an API or Web Service for “high-level” sensors, such as a FitBit or a SNAApp, or a low-level interface for more classic sensors.

In the SOSENS framework, a **Data Parser** entity (usually a software process) is responsible for monitoring, identifying, connecting to and retrieving the sensory data that is appropriate to the social sensing scenario. While most “classic” hardware sensors can be calibrated to detect and store only specific events, social sensors are much more “verbose” and require a data filtering process, performed by this entity.

Afterwards, the filtered data must be converted to RDF that is compatible with the SOSENS environment, using a combination of the SOSENS ontology and domain-specific ontologies. The **Semantic Web Converter** is responsible for this task, in order to create OWL RDF that is ready to be imported to the **Semantic Datastore**. A Semantic Datastore is the “heart” of the SOSENS framework, where all information about sensors and their observations resides. It is a triplestore which supports inferencing and is accessible via standard SW interfaces.

Finally, in order to enhance and augment the collected data (e.g. for Data Quality Assessment or Statistical Analysis), a **Data Processor** entity connects to the Semantic Datastore, retrieves and processes data, updating triples or creating new ones where needed.

### Motivational Scenario, revisited

Given the above, the motivational scenario of Chapter 1 can be described as such:

1. A Data Parser is responsible for detecting and collecting the hardware (digital pedometer) & social (SNS) observations of users about physical activity.
2. The information is forwarded to the Semantic Web Converter, which conforms to the SOSENS ontological standards and creates RDF representations of the input. It converts hardware sensor data to `ssn:Observations`, SNS posts to `sioc:Posts` and `ssn:Observations`, while at the same time tracking provenance where needed via PROV- & SOSENS-related properties & entities.

3. The data is subsequently forwarded to the Semantic Datastore where it is stored and accessible for CRUD<sup>14</sup> operations via its interfaces.
4. A Semantic Processor connects to the Semantic Datastore in order to retrieve physical activity data from multiple sources, compare them to compute quality & trust scores and store the produced information back in the Semantic Datastore.

## 4.2 SOSENS Architecture Building Blocks (ABBs)

In order to implement workflows such as the one presented in the revisited motivational scenario, the SOSENS framework defines *four* building blocks, shown in **Figure 23** by their respective “cog” icon. The *architecture* of these buildings blocks is described in the sections that follow, which focus on design and requirements, rather than actual software solutions. Using this approach, different implementations of the SOSENS framework may use different software solutions, but adhere to the same set of architectural & design principles.

### Please note:

- Some parts of the architecture that are not able to be part of a building block without applying restrictions that would hinder the implementation-agnostic nature of the SOSENS framework are denoted as *non-formative*.
- The *base prefix* (:) refers to the SOSENS ontology namespace
- The *inverse properties*, where available, can be used instead of the properties listed as SOSENS requirements

### 4.2.1 ABB - Data Parser (DPa)



**DATA  
PARSER**

The Data Parser (DPa) Architectural Building Block is a composite structure responsible for *identifying & collecting* the data required for a given Social Sensing scenario. Architecturally, it can encapsulate many functions: Connecting to sensor endpoints, identifying suitable data, retrieving them etc.

---

<sup>14</sup> Create-Read-Update-Delete

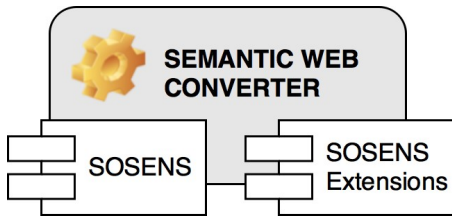


Semantically, it is defined in SOSENS as a `:Parser`, which groups the identifying, retrieval and parsing in one universal entity and one universal process (`:ParsingMethod`), which can further be defined by subclasses and subprocesses if needed. All activities (`:ParsingActivity`) performed by the ABB must be tracked and recorded via PROV & SOSENS sub-properties, as shown in the list of SOSENS Requirements below. It should be noted that converting data to RDF is assigned to another ABB (the Semantic Web Converter), which is presented later in this chapter.

### SOSENS Requirements:

<b>DPa_ABB_R1</b>	<p>The DPa ABB must define one or more methods used to identify, retrieve and parse data via <code>:ParsingMethod</code>.</p> <p>The implementation and semantic definition of the methods is non-formative.</p>
<b>DPa_ABB_R2</b>	<p>The DPa ABB must define one or more <code>:ParsingActivity</code> which retrieve(s) a source entity from a sensor endpoint and converts it to an <code>ssn:Observation</code>. The link to an actual <code>:Source</code> is optional for hardware sensing but must be defined in a social sensor as a <code>sioc:Post</code>.</p> <p>The detailed semantic definition of <code>:Source</code> and <code>:ParsingActivity</code> is non-formative.</p>
<b>DPa_ABB_R3</b>	<p>For each <code>:ParsingActivity</code>, the DPa ABB must provide sufficient SOSENS &amp; PROV metadata using the following properties:</p> <p><code>:hasParsingMethod</code>, <code>prov:startedAt</code>, <code>prov:endedAt</code>, <code>prov:wasAssociatedWith</code>, <code>prov:generated</code></p>
<b>DPa_ABB_R4</b>	<p>For each <code>sioc:Post</code> that is listed as a <code>:Source</code> (see DPa_ABB_R2) the DPa ABB must provide sufficient SIOC metadata using at least the following properties:</p> <p><code>sioc:creator_of</code>, <code>dcterms:created</code>, <code>sioc:content</code>, <code>sioc:has_container</code>, <code>sioc:has_space</code></p>
<b>DPa_ABB_R5</b>	<p>For each user that is the creator of a <code>sioc:Post</code>, the DPa ABB must declare a <code>foaf:Person</code> and its social connections using <code>foaf:knows</code>. Other metadata (<code>foaf:firstName</code> etc.) are optional but recommended.</p>

### 4.2.2 ABB – Semantic Web Converter (SWC)



The Semantic Web Converter (SWC) ABB is a “helper” building block which receives input such as sensor metadata, observations & SOSENS-T ratings and follows the SOSENS ontology in order to convert them to their respective semantic sensing artifacts of Chapter

3. Since this is, essentially, a transformation between data formats (hence the “helper” tag above), the related activities are not semantically tracked by SOSENS. The details & intricacies of choosing the correct data to convert should be declared and tracked via the related methods, such as the `:ParsingMethod` of the DPa ABB.

However, the SWC plays an important role in the SOSENS framework as it is the entity where SOSENS-related **ontological integrity constraints** can be put in place, given that OWL’s Open World Assumption (OWA) makes integrity constraints difficult<sup>15</sup>. For example, a SWC can “refuse” to convert an observation that does not define basic SSN (and SOSENS) data such as a monitored property, an observation result time, or the SIOC post used as a source for a parsing activity. In addition, a SWC can be *decoupled* from the framework and provided separately, to promote Semantic Web data creation that conforms to the SOSENS ontological constructs (as is the case with the SOSENS Web API SBB, presented later).

#### SOSENS Requirements:

<b>SWC_ABB_R1</b>	The SWC ABB must define a list of acceptable inputs that relate to entities in a SOSENS environment (e.g. a <code>ssn:Sensor</code> , a <code>sioct:Post</code> ) and accepted formats for their data (e.g. JSON)
<b>SWC_ABB_R2</b>	The SWC must follow the SOSENS ontology, as well as SOSENS extensions to convert input into appropriate RDF, in a serialisation chosen by the user (e.g. turtle, RDF/XML etc.)
<b>SWC_ABB_R3</b>	The SWC must return errors & debugging information when a SOSENS-related property or instance reference is missing

<sup>15</sup> In [96], Tao et al. make the case for a rule-based integrity constraint (IC) language which, if standardised, may make IC checks easier.

### 4.2.3 ABB – Data Processor (DPr)



#### DATA PROCESSOR

The Data Processor (DPr), is responsible for performing any sort of post-processing on sensor data that exist in a SOSENS environment, such as Data Quality Assessment. These tasks must be linked to a SOSENS extension, which is why the ABB itself is split into subclass-ABBs that define specific, per extension, requirements.

This thesis defines one major SOSENS extension, SOSENS-Trust (see Chapter 3.4.2) and thus presents the SOSENS-Trust Data Processor ABB (abbreviated as ST\_DPr\_ABB). Future work in extending the SOSENS Ontology may result in additional Data Processor ABBs, which should follow the same design methodology.

#### SOSENS-Trust Data Processor ABB

The ST\_DPr\_ABB is semantically defined in SOSENS as a `sosens-t:TrustAuthority` which follows a `sosens-t:RatingMethod` in order to compute a `sosens-t:Rating`. The list of ABB requirements that derive from such workflows can be found below. As was the case with the DPa\_ABB, converting data to RDF is performed in the Semantic Web Converter ABB.

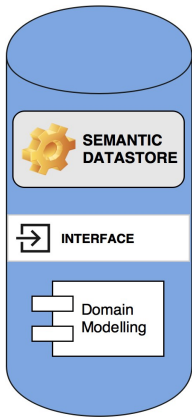
**Note:** For this section only, the base (:) prefix refers to the SOSENS-T namespace

#### SOSENS Requirements:

<b>ST_DPr_ABB_R1</b>	The ST_DPr_ABB must define one or more methods used to retrieve and process SOSENS data using <code>:RatingMethod</code> . The implementation and semantic definition of the methods is non-formative.
<b>ST_DPr_ABB_R2</b>	The ST_DPr_ABB must define one or more <code>:RatingActivity</code> which uses ( <code>prov:used</code> ) one or more SOSENS entities (e.g. <code>ssn:Observation</code> ) in order to compute a <code>sosens-t:Rating</code> . The detailed semantic definition of <code>:RatingActivity</code> is non-formative.
<b>ST_DPr_ABB_R3</b>	For each <code>:RatingActivity</code> , the ST_DPr_ABB must provide sufficient SOSENS-T & PROV metadata using the following

	properties (see Chapter 3.4.2): :hasRatingMethod, prov:startedAt, prov:endedAt, prov:wasAssociatedWith, prov:generated
<b>ST_DPr_ _ABB_R4</b>	For each :Rating, the ST_DPr_ABB must provide a :RatingValue which is linked to a :RatingActivity using :hasRatingValue

#### 4.2.4 ABB - Semantic Datastore (SDS)



The Semantic Datastore ABB is a data storage block responsible for storing all of the information present in a SOSENS environment. It must conform to current SW standards<sup>16</sup>, such as RDF & OWL Inferencing, as well as provide a SPARQL 1.1 query interface.

The SDS ABB is a “classic” Semantic Web RDF triple- or quad-store. A variety of well-known SBBs exist for this ABB as is discussed in the section that follows. Most of these include reasoning (inference) as a built-in function, although it can be added separately. Triplestores should not be confused with

NoSQL/Graph databases, which are more generic, can store different types of graphs but generally do not support inferencing. In SOSENS the Semantic Datastore is also the building block where domain-specific knowledge should be stored, usually during the bootstrapping of a social sensing scenario (see e.g. the pilot of Chapter 5).

#### SOSENS Requirements:

<b>SDS_ABB_R1</b>	The SDS ABB must adhere to the following Semantic Web standards: <ul style="list-style-type: none"> <li>- RDF (<a href="http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/">http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/</a>)</li> <li>- OWL (<a href="http://www.w3.org/TR/owl-ref/">http://www.w3.org/TR/owl-ref/</a>)</li> </ul>
<b>SDS_ABB_R2</b>	The SDS ABB must provide a query interface for SPARQL 1.1

<sup>16</sup> At the time of writing. See [http://www.w3.org/2001/sw/wiki/Main\\_Page](http://www.w3.org/2001/sw/wiki/Main_Page)

## 4.3 Sample SOSENS Solution Building Blocks (SBBs)

As discussed earlier, the point of designing ABBs instead of directly defining SBBs is to allow for further customisation, supporting the implementation-agnostic aspect of the SOSENS framework.

Due to the lack of a uniform way to define `Parsing & Rating` methods - at least in regards to the current status of SOSENS & the Semantic Web in general - some SBBs, such as the `Data Retrieval & Data Processing` ABBs, can be highly application-specific. Advances and future standards such as `Open Social`<sup>17</sup> may lead to more generic SBBs for these categories in the future.

On the other hand, SBBs for the `Semantic Web Converter` & the `Semantic Data Storage` ABBs are abstract enough to be re-usable in almost all SOSENS implementations. The sections that follow present existing solutions and/or solutions custom-built for SOSENS, which satisfy these two ABBs.

### 4.3.1 SBB – Semantic Datastore

As discussed earlier, the SDS ABB is a classic SW triplestore, which supports OWL inferencing and SPARQL. A variety of solutions, both open-source and commercial already exist for storing RDF triples (see **Figure 24**).



Many of these solutions have been used for large-scale real-life applications (e.g. Virtuoso powers the DBpedia SPARQL endpoint<sup>18</sup>, JENA TBD is used in the UK's

<sup>17</sup> Open Social: <http://www.opensocial.org>

<sup>18</sup> <http://dbpedia.org/sparql>

open data website<sup>19</sup> and GraphDB was used by BBC<sup>20</sup>). In addition, some are now described as “Large Triple Stores<sup>21</sup>”, with inferencing support of up to 1 trillion triples (as reported by Oracle<sup>22</sup>).

As defined by the SDS ABB, a triplestore that adheres to the RDF, OWL standards and supports SPARQL can be instantly considered a SOSENS SBB, which creates an “off-the-shelf” approach to bootstrapping and scaffolding social sensing environments. A detailed list of SW triplestores can be found online at the W3C SW Wiki:

<https://www.w3.org/2001/sw/wiki/Tools>

During the design, user testing, evaluation as well as piloting of the framework, the popular Ontotext GraphDB<sup>23</sup> was used.

### 4.3.2 SBB – Semantic Web Converter

Critical to the adoption of Semantic Web technologies anywhere is reducing the *entry barrier* for their implementation. To this effect, a generic Web API was developed during the course of this thesis, entitled the “SOSENS Web API”.

#### The SOSENS Web API

This Web API converts sensory data expressed in the common, easy-to-construct JSON<sup>24</sup> format to their respective SOSENS representation. It is built using the Python programming language and uses Bottle<sup>25</sup> as the HTTP server and RDFLib<sup>26</sup> for converting JSON into the various Semantic Web RDF serialisations (XML-RDF, N3 etc.).

The SOSENS API can help users construct rich semantic representations of their data, without necessarily knowing in-depth how the SOSENS family of ontologies (or SSN) is structured. Each SOSENS API endpoint has a *simple* and *advanced* usage mode, targeted to SW non-experts and experts respectively. This is because

---

<sup>19</sup> <http://data.gov.uk>

<sup>20</sup> <http://ontotext.com/company/customers/bbc-dynamic-semantic-publishing/>

<sup>21</sup> <https://www.w3.org/wiki/LargeTripleStores>

<sup>22</sup> Oracle 1 Trillion RDF benchmark: <http://bit.ly/1l98flk>

<sup>23</sup> <http://ontotext.com/products/graphdb/>

<sup>24</sup> RFC 4627 (JSON) - <https://tools.ietf.org/html/rfc4627>

<sup>25</sup> Bottle: Python Web Framework - <http://bottlepy.org/docs/dev/index.html>

<sup>26</sup> rdflib - A Python library for working with RDF - <https://code.google.com/p/rdflib/>

SSN intentionally does not include common building blocks such as a time and value ontology. To facilitate deployments by non-experts, a *simple mode* request sent to the API uses pre-determined ontologies to describe these, by employing:

1. existing ontologies (such as the W3C's TIME ontology<sup>27</sup>) and
2. ontologies and conventions custom-built for the SOSENS framework, such as the SOSENS Generic SSN Value - see chapter 3.4.1)

Expert users can override these default choices and still take advantage of the JSON-style input to rapidly create SW data.

The API serves the *outgoing* edges/properties of the SOSENS family of ontologies, with incoming edges being inferred by `owl:inverseProperty`. **Figure 25** shows an example of how a Social Sensor's *temperature* observation, expressed in JSON, is converted to its OWL/N3 representation by the SOSENS API. A Hardware Sensor's observation about the same property is also shown for comparison.

It is easy to see that a non-expert user could create the input string without explicit knowledge of SW technologies, as the variables requested represent the “basics” of each sensor observation such as:

- who produced it
- what is it about
- when was it produced

The API provides similar functions for all SOSENS & SOSENS-T constructs (e.g. `sosens-t:Rating`, `sosens-t:TrustAuthority` etc.), conforms to the SWC\_ABB requirements defined in chapter 4.2.2 and is accompanied by detailed documentation which provides a list of accepted inputs. Error messages are returned to the user in the case of missing SOSENS elements. A sample of the API's capabilities and syntax can be found in **Appendix B**.

---

<sup>27</sup> Time Ontology in OWL - <http://www.w3.org/TR/owl-time/>

## POST api/v1/observation

JSON (IN)	<pre> "data": [   Observation   #1 (Social)   {     "observedBy": "Person_1",     "feature": "Area_1",     "value": "Warm",     "valueType": "quality",     "property": "temperature",     "time": "2015-05-20T14:30:00",     "wasParsedFrom": "Facebook_Post_1",     "wasResultOfParsingActivity": "Parsing_Activity_1"   },   Observation   #2 (Hardware)   {     "observedBy": "DigTherm_1",     "feature": "Area_1",     "value": "50",     "valueDataType": "float",     "valueType": "quantity",     "property": "temperature",     "unitOfMeasurement": "Celsius",     "time": "2015-05-20T14:30:00"   } ] </pre>
N3 (OUT)	<pre> :Digtherm_1_observation_temperature_2015_05_20_14_30_00   a ssn:Observation ;   ssn:featureOfInterest :Area_1 ;   ssn:observationResultTime :instant_2015_05_20_14_30_00 ;   ssn:observedBy :Digtherm_1 ;   ssn:observedProperty :temperature ;   ssn:sensorOutput :Digtherm_1_sOut_temperature_2015_05_20_14_30_00 .  :Person_1_observation_temperature_2015_05_20_14_30_00   a ssn:Observation ;   sosens:wasParsedFrom :Facebook_Post_1 ;   sosens:wasResultOfParsingActivity :Parsing_Activity_1 ;   ssn:featureOfInterest :Area_1 ;   ssn:observationResultTime :instant_2015_05_20_14_30_00 ;   ssn:observedBy :Person_1 ;   ssn:observedProperty :temperature ;   ssn:sensorOutput :Person_1_sOut_temperature_2015_05_20_14_30_00 .  :Digtherm_1_oVal_temperature_2015_05_20_14_30_00   a ssn:ObservationValue ;   sosens:hasQuantityUnitOfMeasurement :Celsius ;   sosens:hasQuantityValue "50.0"^^xsd:float .  :Digtherm_1_sOut_temperature_2015_05_20_14_30_00   a ssn:SensorOutput ;   ssn:hasValue :Digtherm_1_oVal_temperature_2015_05_20_14_30_00 .  :Person_1_observationValue_temperature_2015_05_20_14_30_00   a ssn:ObservationValue ;   sosens:hasQualityValue "Warm"^^xsd:string .  :Person_1_sOut_temperature_2015_05_20_14_30_00   a ssn:SensorOutput ;   ssn:hasValue :Person_1_oVal_temperature_2015_05_20_14_30_00 .  :instant_2015_05_20_14_30_00 a time:instant ;   time:inXSDDateTime "2015-05-20T14:30:00"^^xsd:dateTime . </pre>

**Figure 25:** Converting a JSON input from a Hardware Sensor (top) to OWL/N3 (bottom) using the SOSENS API (simple mode).  
Web demo available at <http://phd.pagkalos.com/sosens>



## 4.4 Evaluation

The previous chapters presented the SOSENS framework, its ABBs and some sample SBBs. Two parameters can be considered important for the evaluation of the framework: (1) its contribution to raising overall information quality and (2) usability and performance.

### Reference Implementation

In order to evaluate the SOSENS framework, a SOSENS reference implementation was defined, using:

- the GraphDB SBB
- the SOSENS API SBB presented earlier.

The combination of these two SBBs provides a good starting point for scaffolding social sensing scenarios using SOSENS, as it creates a simple workflow of producing JSON data → converting them to RDF using the SOSENS API and → storing them in GraphDB. This reference implementation was also successfully used in a pilot implementation (presented in Chapter 5), combined with custom DPa & DPr SBBs.

### 4.4.1 Information Quality

Since SOSENS is a unique framework in terms of its scope and its Semantic Web nature, evaluating its contribution to raising overall Information Quality (IQ) should be viewed from the perspective of potential information consumers. In [97], the authors define 15 such quality dimensions (shown in **Figure 26**) which have been shown to appropriately characterise the majority of information consumer expectations. In the context the SOSENS framework, they are, thus, treated as evaluation dimensions.

*Note: “Information Quality”, as described in [97] and in this section, should not be confused with the process of Data Quality Assessment that is discussed throughout this thesis. IQ, in this context, refers to all of the information collected during a social scenario and not just the Data Accuracy & User Trust/Reputation expectations.*

Category	#	Dimension	Definition
<b>Representational</b>	1	Interpretability	[...] information is in appropriate language and units and the information definitions are clear
	2	Ease of understanding	[...] information is clear without ambiguity and easily comprehended
	3	Representation al consistency	[...] information is always presented in the same format and are compatible with previous information
	4	Concise representation	[...] information is compactly represented without being overwhelming (i.e., brief in presentation, yet complete and to the point)
<b>Accessi-bility</b>	5	Accessibility	[...] information is available or easily and quickly retrievable
	6	Access security	[...] access to information can be restricted and hence kept secure
<b>Intrinsic</b>	7	Accuracy	[...] information is correct, reliable and certified free of error
	8	Reputation	[...] information is trusted or highly regarded in terms of their source or content
	9	Believability	[...] information is accepted or regarded as true, real and credible
	10	Objectivity	[...] information is unbiased (unprejudiced) and impartial
<b>Contextual</b>	11	Value-added	[...] information is beneficial and provide advantages from its use
	12	Relevancy	[...] information is applicable and helpful for the task at hand
	13	Timeliness	[...] the age of the information is appropriate for the task at hand
	14	Completeness	[...] information is of sufficient depth, breadth, and scope for the task at hand
	15	Appropriate amount	[...] the quantity and volume of available information is appropriate

**Figure 26:** The 15 most important data quality dimensions from consumer perspective [97]. Order of dimensions slightly altered to align with the relevant textual description of this section.

## How SOSENS satisfies IQ dimensions

Just by using Semantic Web technologies and the SOSENS family of ontologies, the SOSENS framework satisfies all of the *Representational* requirements (#1-4: *Interpretability*, *Ease of Understanding*, *Representational Consistency*, and *Concise Representation*). Using SSN as a base (via the SOSENS ontology) means that data is always in an appropriate language and the units and data definitions are clear. Additionally, due to the structure provided by OWL and the use of the SOSENS API, data collected via SOSENS is always represented in the same format.

In regards to the *Concise Representation* requirement, it can be said that Semantic Web data is meant for machine consumption and thus can be overwhelming at times (even though, these days, N3 and similar notations make them more human-readable than before). Nevertheless, data can be compactly represented by pre-defined SPARQL queries, the results of which are presented to the user via graphical interfaces (e.g. one query to show all Sensor instances, another to show its results, its measuring capabilities etc). In addition, visual SPARQL query builders, such as the ones presented in [98] and [99], can be used over the SOSENS framework interface to provide a better user-experience. This allows the data to be presented both in machine- (OWL/N3) and human-readable (GUI/SPARQL interface) format at the same time.

The use of SPARQL and a triples database makes sure that data is easily and quickly retrievable, as specified by the *Accessibility* (#5) requirement. This is strongly related to *Access Security* (#6), as triples databases and their APIs can (and should) be restricted, and data kept secure (e.g. the Sesame triplestore can be secured via web-server directives). As for the *Intrinsic* category, SOSENS-T strongly satisfies the *Accuracy* (#7), *Reputation* (#8) and *Believability* (#9) requirements, as the SOSENS-T ratings that exist in a SOSENS environment provide the necessary data to satisfy them; Accuracy and Reputation directly, as shown in the example, and Believability through a combination of the two.

Finally, the remaining requirements (#10-15) can be considered application-specific, and although not directly enhanced by SOSENS, the data available and collected by the framework can be used to determine them. For example, Observation Result Times can be used to determine *Timeliness* (#13) and the Measurement Capabilities of each sensor can determine *Relevancy* (#12).

#### 4.4.2 Usability and Performance

It is clear from the above that the SOSENS framework contributes to data quality but it is also critical to ensure that using it does not impact usability or performance in Social Sensing environments. In regards to usability, an important barrier, as mentioned before, is the fact that SW technologies are primarily designed for machine interaction, making reading and writing potentially difficult for human users.

However, as shown earlier, the SOSENS API makes creating SW data much easier. A detailed manual (with examples, such as the ones presented in the SOSENS Web Demo) has aided users in a small focus group (comprised of 3 e-Health professionals<sup>28</sup>) to create semantic SOSENS data quickly, by expressing them in simple JSON strings. Some amount of “manual” work is needed to initially configure the system, such as setting up the domain-specific ontologies and sensor and observation subclasses according to the measured properties (as shown in the Pilot implementation presented in the next chapter), but after this, data entry to a SOSENS system is straightforward and can even be performed by non-experts. Alternatively, one could directly assign the bootstrapping work to a SW expert beforehand.

As for data retrieval and consumption, SPARQL visualisation technologies, as discussed in the previous section can facilitate the process. Pre-configured SPARQL queries such as “show all sensors”, “show all observations from this sensor” and “show all social sensor observations” have been very useful in getting an initial idea of the data in a SOSENS environment.

In regards to the measurable dimensions of the framework, as an example for the performance and response times of using the SOSENS framework to model a Social Sensing space, the response times of a sample group of SOSENS-related procedures was measured in a *commercial* Intel Core i5-3570 machine with 8GB of RAM running Windows 8 (**Figure 27**).

Although the primary focus of this work is the semantic scaffolding of Social Sensing environments, which is, essentially, implementation-agnostic, SW technologies have reached a point where the added value from the rich description of data has little impact, at least on common CRUD operations. With the advent of sophisticated SW software suites such as GraphDB [100] and the constant

---

<sup>28</sup> Users were researchers in the EU e-Health project epSOS (<http://www.epsos.eu>) . Focus group study carried out in December 2015.

optimisation of OWL inference engines, dealing with SW data of the billion-triples scale is easier than before, allowing near real-time results for common queries.

Sample response times for common SOSENS procedures		
Converting Data ( <i>SOSENS API</i> )	Converting 1 SOSENS Observation to RDF	0.05 sec
	Converting 17000 SOSENS Observations to RDF	10 sec
Importing Data ( <i>GraphDB</i> )	Importing 1 SOSENS Observation	Negligible
	Importing 17000 SOSENS Observations	4 sec
Retrieving Data in RDF ( <i>GraphDB</i> )	Retrieving all observations of a single Social Sensor	0.055 sec
	Retrieving 17000 SOSENS observations	0.202 sec
<b>Figure 27:</b> Sample response times for common SOSENS procedures		

## Chapter Conclusions

This chapter presented the SOSENS framework, a *software architecture* for modelling and managing Social Sensing environments based on the SOSENS family of ontologies presented in Chapter 3. To describe the framework, a list of Architecture Building Blocks (ABBs) was presented, which establish requirements to be implemented by Solution Building Blocks (SBBs) in a SOSENS environment.

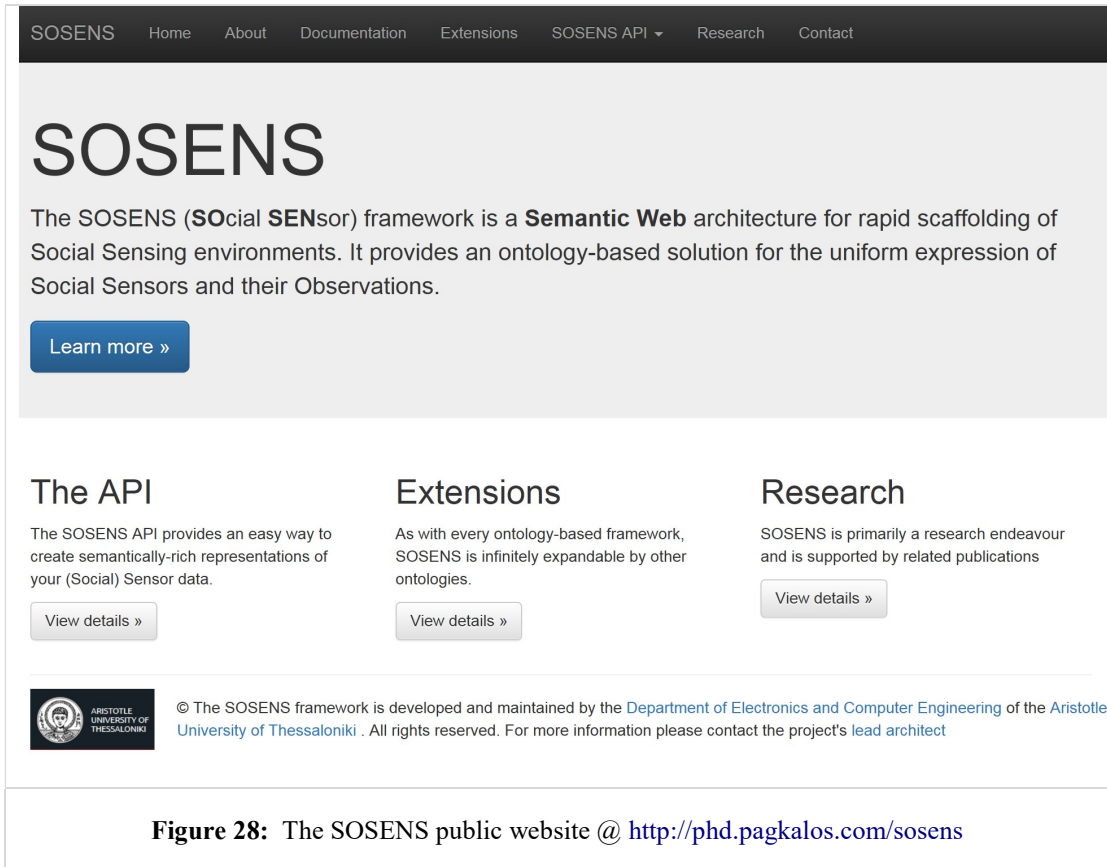
A collection of SOSENS SBBs that conform to the ABB requirements can act as a semantic “scaffold” for heterogeneous sensing environments. Sample SBBs were presented where appropriate, one of which is the SOSENS Web API, which can help users construct rich semantic representations of their data, without necessarily knowing in-depth how the SOSENS family of ontologies (or SSN) is structured.

The SOSENS architecture was evaluated against 15 expected dimensions of information quality and it was shown that it satisfies all requirements and thus can be considered a framework, the application of which confers an increase in overall information quality. Furthermore, employing the framework has a negligible cost in common (e.g. CRUD) procedures as was shown by measuring sample response times using a reference implementation, which is also used in the pilot presented in the chapter that follows.

## Public website

A dedicated website for the SOSENS framework (see **Figure 28**) can be found at:

<http://phd.pagkalos.com/sosens>



**Figure 28:** The SOSENS public website @ <http://phd.pagkalos.com/sosens>

The website acts as the public portal for information regarding the framework: documentation, availability, extensions, research publications etc. In addition, it provides a Web Demo of the SOSENS API which showcases the API's & the framework's capabilities. In the demo, a user may:

- supply the basic characteristics of sensor observations
- create a JSON request string and
- forward the string to a permanent SOSENS API endpoint in order to receive SOSENS RDF.

**Figure 29** shows an example of the above workflow.

Please choose the level of detail of this demo (and the SOSENS API Usage Mode):

Simple Advanced

Please fill in the form below manually or [fill the form with demo values](#)

## SENSOR OBSERVATIONS

[Show Help](#)

Observation

observedBy\* Person\_1

feature\* Area\_1

property\* temperature

time\* 2015-05-20T14:30:00

valueType\* ☐ Quantity ☒ Quality

value\* Warm string

wasParsedFrom Facebook\_Post\_1

wasResultOf Parsing\_Activity\_2

Observation

observedBy\* Person\_1

feature\* Area\_1

property\* temperature

time\* 2015-05-20T14:35:00

valueType\* ☐ Quantity ☒ Quality

value\* Cold string

wasParsedFrom Facebook\_Post\_2

wasResultOf Parsing\_Activity\_2

+ Add New Observation CREATE REQUEST

his is the request string as it should be sent to a SOSENS API endpoint. You can POST it yourself (please [contact us](#) for an API key) or press the "SEND REQUEST TO this web server's SOSENS API"

REQUEST

```

1 {
2   "data": [
3     {
4       "observedBy": "Person_1",
5       "feature": "Area_1",
6       "value": "Warm",
7       "valueDataType": "string",
8       "valueType": "quality",
9       "property": "temperature",
10      "time": "2015-05-20T14:30:00",
11      "wasParsedFrom": "Facebook_Post_1",
12      "wasResultOfParsingActivity": "Parsing_Activity_2"
13    },
14    {
15      "observedBy": "Person 1",

```

SEND REQUEST TO API

RESPONSE

```

1 @prefix dul: <http://www.loa-cnr.it/ontologies/DUL.owl#> .
2 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
3 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
4 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5 @prefix sosens: <http://pagkalos.com/sw/sosens#> .
6 @prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#> .
7 @prefix time: <http://www.w3.org/2006/time#> .
8 @prefix xml: <http://www.w3.org/XML/1998/namespace> .
9 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
10
11 <Digital_Thermometer_1_observation_temperature_2015_05_20_14_30_00> a ssn:Observation ;
12   ssn:featureOfInterest <Area_1> ;
13   ssn:observationResultTime <instant_2015_05_20_14_30_00> ;
14   ssn:observedBy <Digital_Thermometer_1> ;
15   ssn:observedProperty <temperature> ;

```

**Figure 29:** The SOSENS public website – API Demo





# 5 Pilot Implementation

## Chapter Introduction

In order to validate the SOSENS framework in real-life conditions and prove its applicability and practical benefits, a full-scale Social Sensing pilot study was designed, implemented and enhanced using the SOSENS Reference Implementation.

This chapter initially presents the methodology of the NutriHeAl Exercise Social Sensing (ESS) pilot on Facebook, where a group of users self-reported their daily activities using a custom-built Facebook application. SOSENS is used to express Facebook users as Social Sensors and the user-generated content as sensor observations, while at the same time capturing and describing provenance.

Following the pilot's Data Quality Assessment (DQA) methodology, social sensor observations were compared to data from a Fitbit Digital pedometer. Observation quality and social sensor trust was computed using a combination of Fuzzy membership & multinomial Bayesian Reputation systems. SOSENS-Trust is used to model the data & processes in semantically-rich, easily re-usable RDF. The chapter presents how the entirety of the DQA methodology applied in this pilot, as well as its results can easily be re-used in other social sensing scenarios.

Finally, the key advantages of using the SOSENS framework in this Social Sensing environment are discussed at the end of this chapter.

## 5.1 The “NutriHeAl Exercise Social Sensing” pilot

The process of collecting and managing self-reported exercise data is essential in many health disciplines. For example, physical activity diaries are, in conjunction with food diaries, one of the most important tools in gathering patient data in Nutrition and Dietetics [5], [6]. Such subjective methods (“direct observations, diaries, activity logs, recall and questionnaires”) are popular methods for quantifying a selected variable (e.g. physical activity) due to their relatively low cost and the added value of contextual information provided by the user [7]. This is especially true in large-scale studies, where cost and ease of deployment can become a very important factor in the overall success and results of the study.

Using Web and Internet technologies, many of these methods can be digitized and provided online in various Web spaces, such as Social Networking Sites (SNS), which have known increasing popularity over the last years. Facebook, specifically,

has recently reached a phenomenal 1.5 billion+ monthly active users as of Q3 2015 [101]. As expected, the popular website's penetration is also high in young adult groups across the world (e.g. 87% in the US, ages 18-29 [102]). With such a high volume of users, it is natural to consider using SNSs as platforms for a variety of e-Health applications [103] such as activity monitoring. This pool of pre-existing users (people that already use Facebook for their own purposes) allows for a larger target group for non-specific monitoring scenarios or, in the case of pre-determined users, a high chance that some of the users are already on the platform, simplifying account management, link/questionnaire distribution and other similar issues.

As mentioned before, data on SNSs is mostly unstructured and usually provided for different purposes. Social Networking Applications (“SNApps”) are applications, usually Web-based or mobile-phone-based, that are linked to SNSs and can guide a user into providing structured data while, at the same time, making some of the users' social data available. The application's developer can collect the users' input as well as parts of the accompanying social data.



The “**NutriHeAl**”<sup>29</sup> (**Nutrition – Health Alliance**) project (2012-2015) was a nutrition & lifestyle intervention project targeted to Greek municipalities. The project's ultimate goal was intervening to promote a healthier lifestyle using established tools such as personalised nutritional guidance. Given that assessing and monitoring energy balance is very important to such interventions, it also included a *research pilot study for exercise social sensing* as a separate workpackage, led by this thesis' author. The focus of this particular pilot study (the “NutriHeAl Exercise Social Sensing pilot”, herein the “**NutriHeAl ESS pilot**”) was exploring the use of Social Media for exercise monitoring using a calendar-based SNApp.

During the course of the ESS pilot, a custom-built SNApp on Facebook (a “Facebook app”) was developed and tested in a group of young adults in order to monitor their physical activity over time using Facebook. Furthermore, a methodology to assess data quality (where possible) was developed using Digital Pedometer data. As a result, a considerable amount of valuable data was created, which can be of use to many e-Health disciplines. The SOSENS framework was employed in order to make data & processes reusable for humans and machines alike.

---

<sup>29</sup> NutriHeAl Project: <http://www.nutriheal.gr>, funded by the 2007-2013 Greek National Strategic Reference Framework (NSRF)

The NutriHeAl ESS pilot study's usefulness was fully acknowledged by the field's health professionals as it presents the first related research effort that uses a calendar as a Facebook app to self-report exercise (among other activities) in detail. In addition, although digital pedometers step counts have been used in the past as a self-reported variable, in this study they are automatically retrieved (thus there is no room for self-reporting error) and are also used for validating the self-reported exercises, creating a trove of useful data.

The remainder of this chapter will focus mainly on the Computer Science & Semantic Web aspects of the ESS pilot and its implementation via SOSENS – Appendix A provides more information in regards to the e-Health aspects of the pilot & its results.

### 5.1.1 The NutriHeAl Facebook Application

The quintessential component of the ESS pilot was the NutriHeAl Facebook app. Built for the purposes of the NutriHeAl project and based, in part, on previous work by this thesis' author ([104],[105],[106]), the app is entitled “NutriHeAl Activity Diary”. It was accessible publicly through Facebook for the duration of the specific project workpackage and requires a valid Facebook account to use. The app was designed and launched in the Greek language but there are plans to provide full translation packages, as the application has an abstract design approach that can be used in similar research projects. The screenshots presented hereafter use a beta version of the English translation package.

Technically, the app is a W3C standards-compliant Website (built with HTML, PHP and frameworks like Bootstrap & jQuery<sup>30</sup>) that is hosted on a private server and “served” through the Facebook canvas. This allows it to use its own design as well as store its own data, while at the same time benefiting from the Facebook environment integration. What this means, in practice, is that users who click on a link to use the app (e.g. from a Facebook news post, or a post in a Facebook group) “stay” in the Facebook environment which allows them to use all the Facebook services (chat, notifications etc.) while at the same time accessing the application. This also helps to motivate users to use the app while on Facebook, as they do not feel like they have to leave Facebook and stop what they're doing to do so.

When using the application for the first time, users are presented with a Facebook-controlled mandatory dialog which allows them to either accept or deny the permissions required by the app. Apart from the standard public profile data, the

---

<sup>30</sup> Bootstrap: <http://getbootstrap.com>, jQuery: <http://jquery.com>

NutriHeAl app only requests the user's *list of friends*, which –as discussed later– can be used for a multitude of purposes. The app's privacy policy (according to Facebook Policy<sup>31</sup>) explicitly stated that social connection data may be used anonymously for further research and social network analysis.

After authorizing the app, users are presented with the app's homepage. The data-collection methodology is based on a two-step approach, visualized through a tabbed interface. The first tab (Profile) collects the user's basic information and the second tab (Activity Diary) contains a weekly calendar where users can add their daily activity. The third tab (Results) requires no user input and shows result graphs that combine Fitbit & app data. The user interface is described in detail below.

**NUTRIHEAL**

Profile Activity Diary

Hello ! Please check that everything is in order and press "Next"

Name John Gender ☒ Male ☐ Female

Surname Test Age - 28 +

E-mail test@test.com Height - 180 cm +

Occupation Nutritionist Weight - 79 kg +

Your **Basal Metabolic Rate (BMR)** is the minimum amount of energy your body needs to function ?

BMR 1882 kcal/day 78.4 kcal/hour

Your **Body Mass Index (BMI)** is a measure of relative weight based on your weight and height ?

BMI 24.4 Normal Weight

**Figure 30:** The NutriHeAl Facebook app – Profile Tab

## Profile Tab

In the Profile tab (**Figure 30**), users enter their Name, Surname, E-mail and Occupation as well as their Gender, Age, Height and Weight (self-reported). The app uses these to automatically calculate the user's BMI, BMR and BMR/hour (using the Schofield equation [107]), briefly explain what they are and provide

<sup>31</sup> Facebook policy: <https://developers.facebook.com/policy>

feedback in the case of BMI (using the BMI classification as established in the WHO 2000 report [108]). By hovering over the “?” icon, the user can get more information in regards to these metrics. Both the self-reported data (e.g. Age, Weight, and Height) and the data calculated by the app (BMI, BMR) are saved in a MySQL database when the user proceeds to the next tab. As mentioned before, this database is not related to Facebook in any way and is stored in a separate, secure environment.

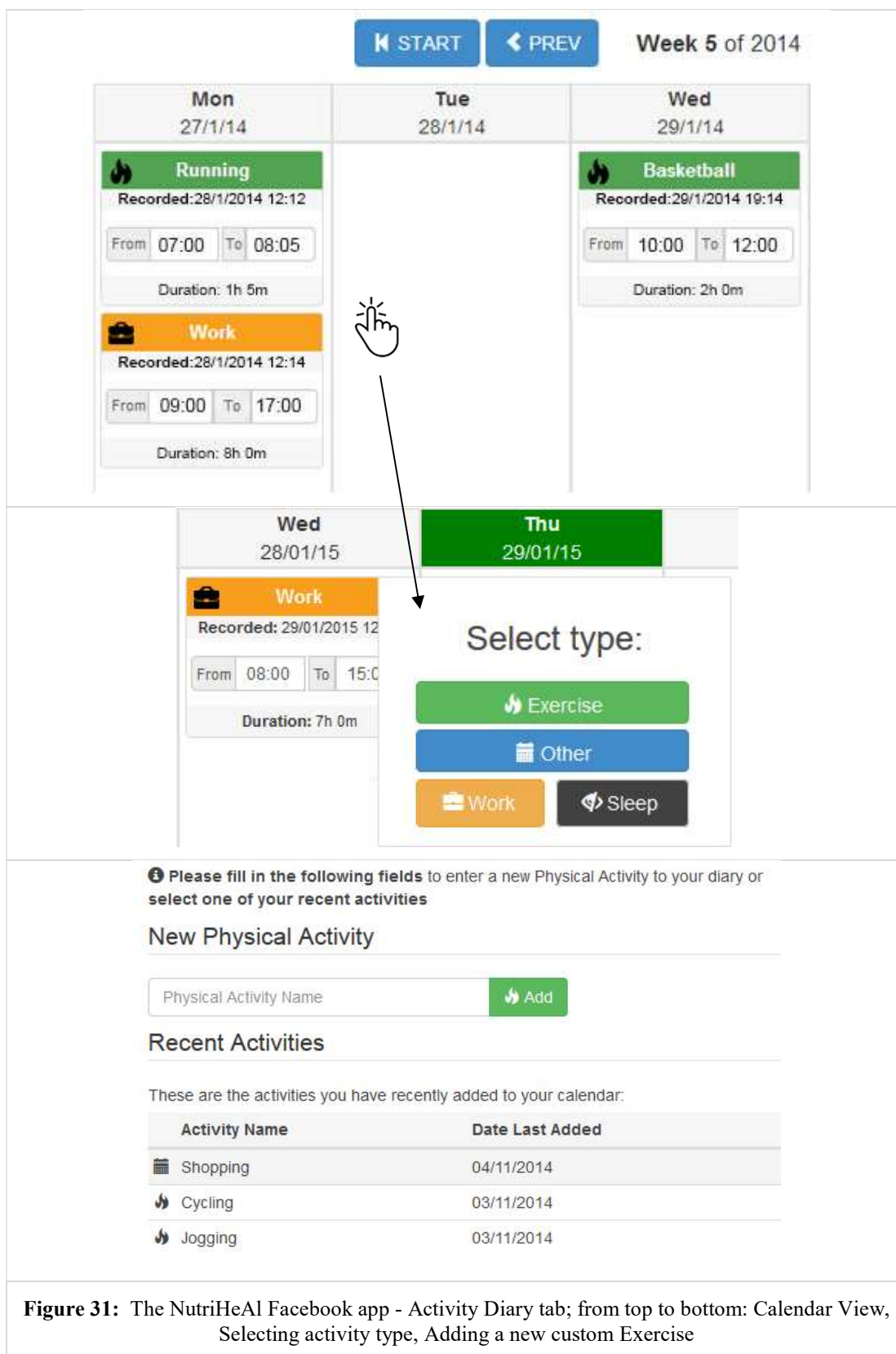
### Activity Diary Tab

The Activity diary tab is an interface that resembles a weekly calendar (**Figure 31**, top section), where users are able to add a new activity by double-clicking on the empty “white space” of each day, as denoted by the click (👉) icon in the figure. The users are then presented with a dialog (**Figure 31**, middle section) which allows them to either:

- select one of two pre-defined activities – “*Sleep*” and “*Work*”
- add an “*Exercise*” activity (e.g. “Walking”, “Running”, typed in by the user)
- add an “*Other*” activity (e.g. Shopping, Sweeping) activity

In order for users to record activities in a detailed manner, one approach is to utilise a pre-populated activity table for exercise entry such as the well-known Compendium of Physical Activities [109]. In pre-pilot tests with a focus group of 5 participants, it was quickly established that users have difficulty finding and selecting the correct exercise without the presence of an expert. Users would ordinarily miscalculate their walking or jogging speed, select the wrong type of stationary Bicycle activity when in the gym etc. This is an expected outcome and not an issue with the Compendium and related tables, as they are designed for a different purpose (e.g. comparing MET values between activities).

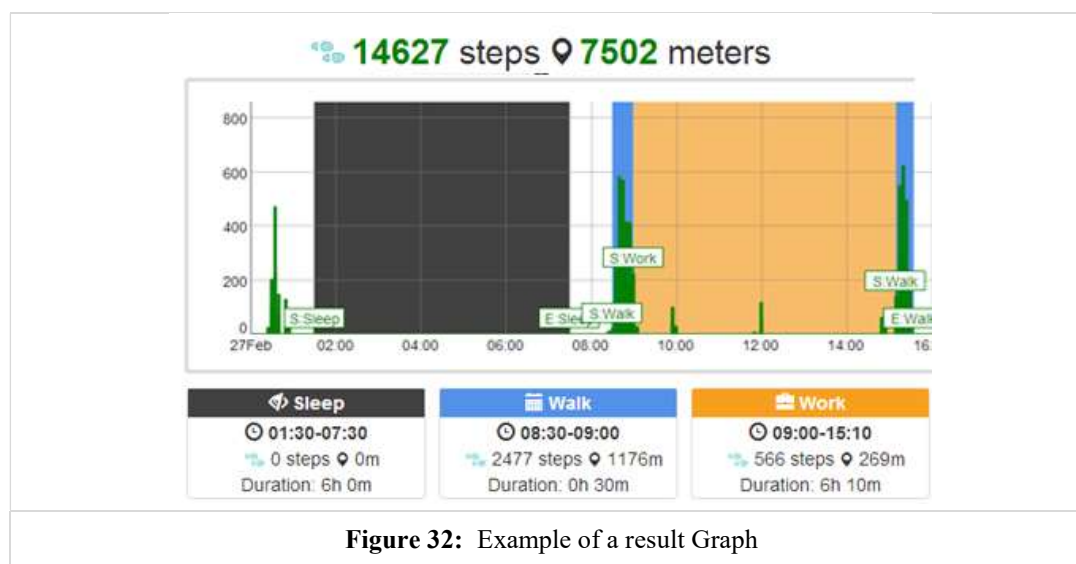
An alternative approach, which was utilised in the NutriHeAl app, is to allow users to specify most of the activities themselves as “free text”. In addition, the system keeps a record of past activities and allows users to quickly re-add them (**Figure 31**, Bottom, “Recent Activities”). Using this approach, the focus group displayed a very low (5-6 sec) turnaround time when adding a new activity to their calendar which also makes it easier for users to enter their data often.



**Figure 31:** The NutriHeAl Facebook app - Activity Diary tab; from top to bottom: Calendar View, Selecting activity type, Adding a new custom Exercise

## Results Tab

These personalised graphs, an example of which can be seen in **Figure 32**, show the user's reported activities as an overlay to recorded digital pedometer data and were accessible only by the specific participant and the overseeing nutritionist (using a username/password combination). These graphs can be used in multiple ways, such as an incentive for users to submit exercise data in order to later view their relation to the steps recorded by the Fitbit. This is a similar approach to what many of the digital pedometer mobile phone apps (incl. Fitbit) can do, but has the added advantage of using the same interface & environment as the pilot study app.



**Figure 32:** Example of a result Graph

## 5.2 Data Collection

### 5.2.1 Pilot Sampling

Users that were part of the NutriHeAl project control population were approached via e-mail and asked “to participate in a separate pilot study for exercise monitoring”. Apart from not being physically active (i.e. not having a permanent or temporary condition that prevents physical activity), the only exclusion criterion was not owning a Facebook account, or lack of willingness to create one for this study. Following a 2-week recruitment period, a randomly sampled group of **49 Greek young adults (n=49, Avg. Age = 24 ± 7, Avg. BMI 22.5 ± 3)**, was assembled. Out of the 49 participants, only 1 did not have a Facebook account and decided to create one for the pilot.

## 5.2.2 Pilot Methodology

Participants were instructed to use the NutriHeAl app on a *daily* basis, for a period of 5 weeks. They were asked to *record their exercise activities*, and use the other category types of the app (“Sleep”, “Work”, and “Other”) only if desired. In addition, participants were provided with a digital pedometer (Fitbit Zip<sup>32</sup>) to wear all day, removing it only when in the shower or engaged in water-based physical activity (swimming etc.). The device was worn in the manufacturer-approved body locations<sup>33</sup> (belt, pants pocket, bra etc., see **Figure 33**). Participants were also provided with an accompanying USB adapter that, via the Fitbit software, automatically uploads data from the pedometer to a central server.



No specific training sessions took place (e.g. teaching users “what activities are considered exercise”), apart from an introductory 1-hour session on how the app works (in terms of the web interface) and how to set-up the Fitbit software on PCs.

To encourage participation and maintain the users’ interest, result graphs which combined Fitbit data with self-reported activity times were released on a participant basis at the end of each week, provided that users have submitted a filled-in weekly activity diary.

In summary, the data collected during the course of the pilot per user were:

- A list of self-reported user activities with time-stamped beginning and end.
- Fitbit Zip data (both as overall steps/day and as specific steps/min every 1 minute, uploaded by the users automatically and stored centrally on the Fitbit servers)
- Any Social data provided by the app (e.g. public profile data, Facebook friends)

<sup>32</sup> <https://www.fitbit.com/zip>

<sup>33</sup> [http://help.fitbit.com/articles/en\\_US/Help\\_article/How-do-I-wear-my-Zip](http://help.fitbit.com/articles/en_US/Help_article/How-do-I-wear-my-Zip)



## 5.3 Using SOSENS to model collected data

The previous sections described in detail the sampling & data collection processes of the NutriHeAl ESS Pilot. Without any sort of semantic mark-up, all of the data, although very useful to a number of health sciences, would be expressed in an application-specific manner, with little to no provenance details. This chapter presents how easy it is to create semantically-rich representations of the data using the SOSENS Framework, for the Social & Hardware sensor observations alike.

### 5.3.1 Bootstrapping

As discussed in chapter 4.2.2, a *bootstrapping* procedure is typically required for each social sensing scenario modelled by SOSENS. The first step of this procedure is choosing the way to model the domain-specific characteristics of the sensing environment. Sometimes, this can be as simple as choosing one of the publicly-available ontologies on the Web. Alternatively, ontologies or specific properties/individuals will need to be created, either to fit the sensing scenario in the best way possible or to quickly bootstrap a project, as was done in the case of the NutriHeAl ESS pilot. To model the domain-specific characteristics of activity monitoring, a simple OWL property (`:PhysicalActivity`<sup>34</sup>) was created and declared as a `subProperty` of `ssn:Property`.

```
:PhysicalActivity a ssn:Property .
```

*RDF/Turtle*

In this specific ontological view, the physical activity performed by users (i.e. the value of the property) is expressed, using the SOSENS Generic SSN Value ontology (see **Chapter 3.4.1**), in:

- Qualitative values (e.g. “walking” or “jogging”), as observed by the user, or
- Quantitative values (e.g. 500 “steps” or accelerometer readings), as observed by hardware sensors<sup>35</sup>.

<sup>34</sup> For the remainder of the document, the base prefix (`:`) refers to <http://pagkalos.com/sw/nutriheal#>

<sup>35</sup> This is by no means a “complete” solution, but rather a solution that fits the pilot study. In the future, it is worthwhile to explore using detailed physical activity ontologies to describe recorded activities on a named instance level such as SMASH (<http://purl.bioontology.org/ontology/SMASHPHYSICAL>) but work done in that space is, at the time of writing, limited.

A second step is creating “helper” OWL subclasses in order to avoid declaring some of the properties of sensors & observations many times. Although optional, this step is not only a time-saver for data-entry but also *reduces the number of triples* in the database, as OWL inference can be used instead. In the NutriHeAl ESS pilot, a “helper” subclass was created for physical activity observations (i.e. observations that have `:PhysicalActivity` as their monitored property, using OWL restrictions as shown in the snippet below:

```
:PhysicalActivityObservation a owl:Class ;
  rdfs:subClassOf ssn:Observation ;
  owl:equivalentClass [
    a owl:Restriction ;
    owl:hasValue :PhysicalActivity ;
    owl:onProperty ssn:observedProperty ;
  ]
.
```

The same methodology was followed for creating a physical activity sensor subclass and a class to describe Fitbits as such sensors:

```
:PhysicalActivitySensor a owl:Class ;
  rdfs:subClassOf ssn:SensingDevice ;
  owl:equivalentClass [a owl:Restriction ;
    owl:hasValue :PhysicalActivity ;
    owl:onProperty ssn:observes ;
  ] .

:FitbitZip a owl:Class ;
  rdfs:subClassOf :PhysicalActivitySensor .
```

Finally, a helper class for expressing values in steps was also created:

```
:UnitOfMeasurement a owl:Class ;
  rdfs:subClassOf owl:Thing .

:Steps a:UnitOfMeasurement .

:stepsValue a owl:Class ;
  rdfs:subClassOf sosens:QuantityObservationValue ;
  owl:equivalentClass [
    a owl:Restriction ;
    owl:hasValue :Steps ;
    owl:onProperty sosens:hasQuantityUnitOfMeasurement ;
  ] .
```

### 5.3.2 Modelling Hardware Sensors (Fitbit Zip) in SOSENS

Since SOSENS is based on the SSN ontology, expressing Fitbit Zip data in RDF is very straightforward. Each Zip was declared as an instance of the Fitbit class and was the creator (via `ssn:observedBy`) of one or more physical activity observations about their respective owner (`ssn:featureOfInterest`). Using the domain-specific information presented above, each observation had a SOSENS Quantity Value of the `xsd:double` datatype (e.g. `5e+02` when the Fitbit zip reported 500 steps / 5 min, as shown in the example RDF below:

```
:FitbitZip_1 a :FitbitZip;

:FitbitZip_1_obs_2014_01_29_07_00_00
  a :PhysicalActivityObservation ;
  ssn:featureOfInterest :JSmith ;
  ssn:observationResultTime :instant_2014_01_29_07_00_00 ;
  ssn:observedBy :FitbitZip_1 ;
  ssn:sensorOutput :FitbitZip_1_SO_PAO_2014_01_29_07_00_00
.

:FitbitZip_1_SO_PAO_2014_01_29_07_00_00
  a ssn:SensorOutput ;
  ssn:hasValue :FitbitZip_1_OV_PAO_2014_01_29_07_00_00 .

:FitbitZip_1_OV_PAO_2014_01_29_07_00_00
  a :StepsValue ;
  sosens:hasQuantityValue 5e+02 .

:instant_2014_01_29_07_00_00 a time:instant ;
  time:inXSDDateTime "2014-01-29T07:00:00"^^xsd:dateTime .
```

### 5.3.3 Modelling Social Sensors (NutriHeAl App) in SOSENS

As discussed in the beginning of this chapter, using an app instead of data mining Facebook wall posts was an intentional design choice for this project, in order to facilitate observation identification, collection and parsing. An app is able to define how to store and export its data which, can more easily be converted to SOSENS observations. The section that follows shows how SOSENS can be used to treat NutriHeAl users as Social Sensors and seamlessly integrate their observations with those of the Fitbits. A graphical representation of the information below can be seen in **Figure 34**):

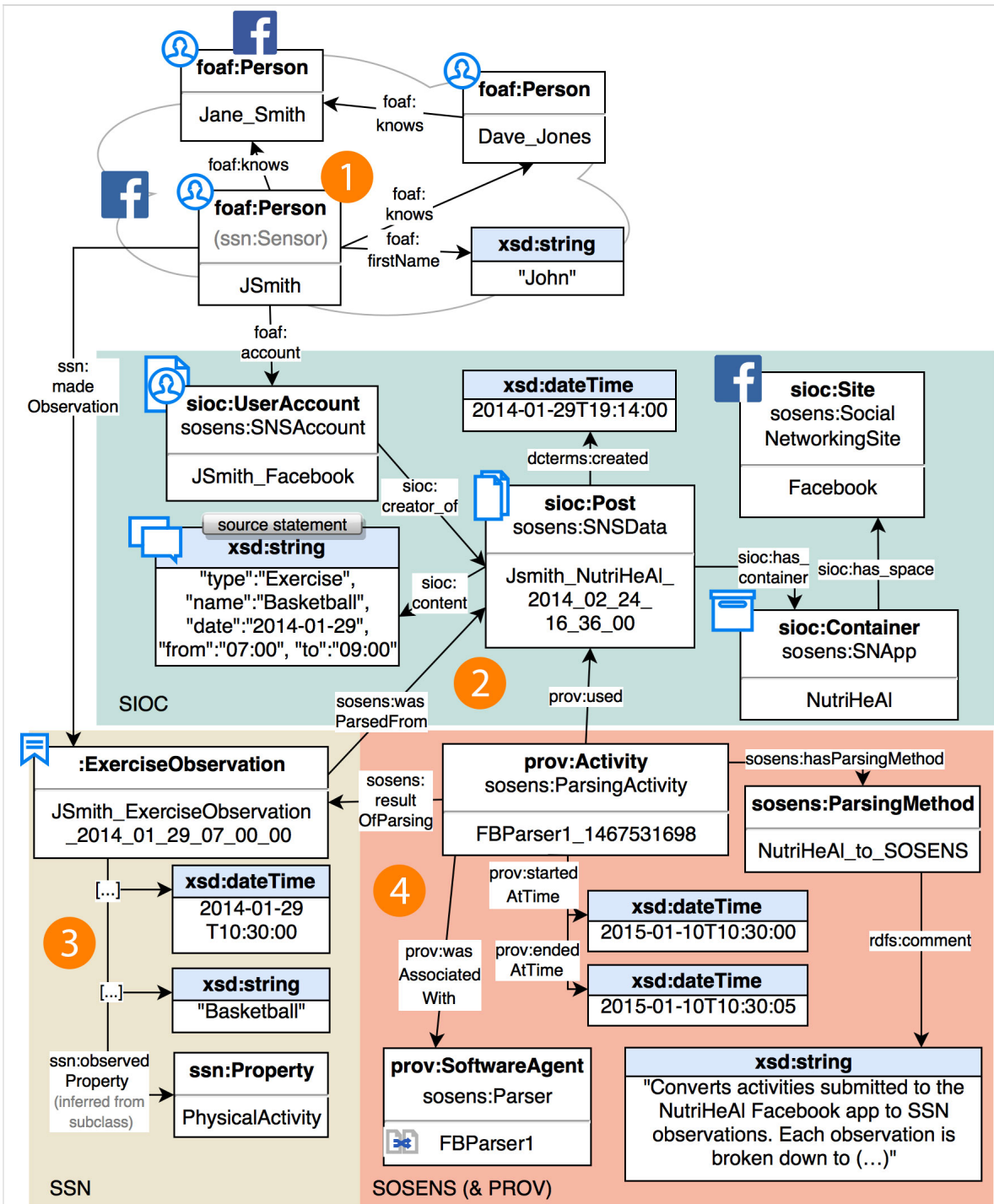


Figure 34: Modelling a Social Observation from the ESS pilot using SOSENS

- 1 **Users:** Each participant was represented by the basic `foaf:Person` class, as is the norm in most Semantic Web systems. The users' details were expressed in common FOAF properties and their friendship network was expressed using the `FOAF : knows` property, as shown in the example below:

```
:JSmith
a foaf:Person ;
foaf:account :Jsmith_Facebook ;
foaf:firstName "John"^^xsd:string ;
foaf:lastName "Smith"^^xsd:string ;
foaf:knows :Dave_Jones ;
foaf:knows :Jane_Smith .
```

RDF/Turtle

- 2 **Source Post:** Each `foaf:Person` was the creator of one or more `SNSData` instances, which is the SOSENS-introduced subclass of a `sioc:Post`. Each post was “contained” in the NutriHeAl SNAApp, which itself is part of the Facebook SocialNetworking Site.

```
:NutriHeAl a sosens:SNAApp ;
sioc:has_space :Facebook ;
rdfs:comment "The Facebook app of the NutriHeAl
Project"^^xsd:string .

:Facebook a sosens:SocialNetworkingSite ;
rdfs:comment "The Social Networking Site \"Facebook\" at
www.facebook.com"^^xsd:string .
```

RDF/Turtle



```
:Jsmith_NutriHeAl_2014_02_24_16_36_00
a sosens:SNSData ;
dct:created "2014-02-24T16:36:00" ;
sioc:content
  "\"type\": \"PhysicalActivity\",
  \"name\": \"Basketball\",
  \"date\": \"2014-01-29\",
  \"from\": \"07:00\",
  \"to\": \"09:00\""^^xsd:string .
```

- 3 **Observations:** Each person was the creator of one or more physical activity observations which were assigned a subclass as defined by the user (e.g. exercise observation, work observation) and the user itself as the feature of interest. When OWL inference is applied, this automatically classifies them as a `ssn:Sensor` as well, due to the `ssn:observedBy` property's `rdfs:domain`.

The (qualitative) value of each observation was an `XSD:String` value, equal to the activity name as it appeared on the NutriHeAl app. Sensor output and observation result times were implemented using the SOSENS defaults (Generic SSN Value and the W3C time ontology, respectively)

**4 Observation parsing & provenance:** One of the basic differences between NutriHeAl app observations and Fitbit Zip observations, apart from the obvious qualitative (textual) description of values instead of quantitative, is that the former are expressed as a time period (“was observed from: `time1` to `time2`”) whereas the latter are have a more standard sensor timestamp format (“was observed on `time`”). Thus, in order to align with Fitbit Zip data, each submitted activity from the NutriHeAl app had to be broken down to 1 observation per 5-minute interval (the equivalent of the social sensor observing the same value every 5 minutes). Conversely, Each NutriHeAl observation was the source of multiple Social Sensor observations.

As mentioned before, this is one of the reasons why the parsing & provenance parts of SOSENS are very important. Apart from making sure that the respective properties are used correctly, this information was declared on the parsing method used, as shown in the code snippet below. The parsing method was declared as a text comment (`rdfs:comment`), although it is possible to describe it in more detail using a custom ontology.

```
:NutriHeAl_to_SSN a sosens:ParsingMethod ;
    rdfs:comment "Converts activities submitted to the
NutriHeAl Facebook app to SSN observations. Each
observation is broken down to (...)"^^xsd:string .
.
```

To link back to the original post, all observations were declared as parsed using this method from a NutriHeAl app `SNSData` item (see **Figure 34**). An example of a full Social Sensor observation can be seen below:

```

:JSmith_ExerciseObservation_2014_01_29_07_00_00      RDF/Turtle
  a :ExerciseObservation ;
  sosens:wasParsedFrom :Jsmith_NHeAl_2014_02_24_16_36_00 ;
  sosens:wasResultOfParsingActivity :FBParser1_1467531698;
  ssn:featureOfInterest :Jsmith ;
  ssn:observationResultTime :instant_2014_01_29_10_00_00 ;
  ssn:observedBy :JSmith ;
  ssn:sensorOutput :JSmith_SO_ExObs_2014_01_29_10_00_00 .

:JSmith_SO_ExObs_2014_01_29_10_00_00
  a ssn:SensorOutput ;
  ssn:hasValue :JSmith_OV_ExObs_2014_01_29_10_00_00 .

:JSmith_OV_ExObs_2014_01_29_10_00_00
  a ssn:ObservationValue ;
  sosens:hasQualityValue "Basketball"^^xsd:string .

:instant_2014_01_29_10_00_00
  a time:instant ;
  time:inXSDDateTime "2014-01-29T10:00:00" .

```

### 5.3.4 Software Implementation

The pilot used the SOSENS Reference Implementation (see Chapter 4.4) which meant that, at this stage of the pilot (data collection), the only customisation needed in terms of SBBs was building a Data Parser (DPa) module.

Given the ESS pilot's design, this was a simple endeavour:

For the *Fitbit observations*, a Python script queried the Fitbit server in pre-determined intervals using the Fitbit API<sup>36</sup> and retrieved the data for each sensor. Given an association between each user and their Fitbit that was provided by each user in the NutriHeAl app's Profile tab and subsequently stored in the app's database, the script created a JSON string such as the one shown below for each sensor observation. The raw data source (the Fitbit output) was not semantically described, as it was simply a transformation between data formats (which is one of the reasons why **DPa\_ABB\_R2** defines the source in hardware observations as optional – see Chapter 4.2.1)

---

<sup>36</sup> <http://api.fitbit.com>

```

"data": [{
  "observedBy": "FitbitZip_1",
  "feature": "JSmith",
  "value": "500",
  "valueDataType": "double",
  "valueType": "quantity",
  "observationClass": "PhysicalActivityObservation",
  "time": "2014-01-29T07:00:00",
  "observationValueClass": "StepsValue"
}]

```

JSON

For the *NutriHeAl Social Sensor observations*, a Python script (semantically described in the previous chapter as the `FBParser1` instance of the `sosens:Parser` class) queried the NutriHeAl app database for new user submissions and created JSON strings for each observation according to the type of physical activity reported, such as the one shown below:

```

"data": [{
  "observedBy": "JSmith",
  "feature": "Jsmith",
  "value": "Basketball",
  "valueDataType": "string",
  "valueType": "quality",
  "observationClass": "ExerciseObservation",
  "time": "2014-01-29T10:00:00",
  "wasParsedFrom": "Jsmith_NH_2014_02_24_16_36_00",
  "wasResultOfParsingActivity": "FBParser1_1467531698" }]

```

JSON

This information was forwarded to the SOSENS API (the Semantic Parsing SBB) in order to be converted to SOSENS-appropriate RDF where it was also assigned a `ParsingActivity` instance, where needed. Finally, data was uploaded to the ESS pilot's GraphDB (the Semantic Data Storage SBB) using SPARQL UPDATE queries via GraphDB's SPARQL interface.

“Filtering” the pilot's data through SOSENS has raised information quality by effectively integrating two heterogeneous data sources and providing the metadata & provenance information needed for decision making. Accompanied by the information presented in the chapter that follows, using SOSENS leads to a number of benefits, further discussed in **Chapter 5.5**.



## 5.4 Data Quality Assessment (DQA)

The previous chapter described the data collection process in the ESS pilot and its implementation in SOSENS. As discussed in **Chapter 3.4.2**, it is important for Social Sensing observations to be accompanied by some sort of data quality indicator, which is why the ESS pilot also included a Data Quality Assessment (DQA) methodology.

Even though the NutriHeAl app can collect data about many types of activities, the focus of the ESS pilot was on collecting and evaluating the self-reported *exercise* activities provided by the users (the social sensors). Apart from being an important type of activity to monitor for lifestyle intervention projects such as this, it is also easily comparable to ground truth from hardware sensors. Thus, by treating the *Fitbits* worn by the users as *ground truth*, one can measure the “closeness” of a social sensor’s statements to that truth. In addition, these values can contribute, over time, to building a social sensor’s (good or bad) reputation as a data provider.

There were *two* steps in the ESS DQA methodology which are discussed in the chapters that follow. First, individual social sensor observations were assigned a quality rating using custom, application-specific metrics. Subsequently, these ratings were aggregated in order to compute a social sensor’s trust.

Although DQA in an ESS scenario is an interesting topic that presents many scientific questions, it should be noted that the focus of this thesis is on using SOSENS to create the re-usable semantically-rich RDF representations of the procedures and the resulting values. As such, the presentation and discussion of the methodology followed in each step is limited to basic information only.

### 5.4.1 DQA - Social Sensor Observation Quality

#### Pilot Methodology

The first step towards DQA in such a scenario is determining quality on an observation basis. In this respect, the exercises reported by the users over the duration of the pilot can be compared to the data provided by the Fitbits, which have been shown to be valid for estimating steps during physical activity ([110]–[112]) and thus can be treated as *ground truth*. Consequently, a conclusion about an observation’s quality can be reached by comparing the two sources over the same duration. For example, when a user reports an activity in the time frame of 13:00-14:00, there is a respective value in steps/min recorded by the Fitbit for that specific duration. Let that value be defined as  $s_{pm}$ .

In collaboration with an expert, exercises in the NutriHeAl ESS pilot were broken down into two different categories according to intensity: **Low+** and **Moderate+**. This categorisation was performed in a semi-unsupervised way: First, a script assigned one of the two pre-defined categories to “common” activities like walking and running in accordance to their MET (METabolic equivalent [109]) value: 3+ MET for Moderate+ and 2+ for Low+. For activities that were not categorised automatically via simple text-matching (387 activities, ~25%), the expert assigned a category manually. **Table 2** shows a sample of the activities submitted by the users, their assigned category and the respective  $s_{pm}$  values.

Name	Type	Duration (min)	Steps	$s_{pm}$
Dancing	Moderate+	45	4527	100.6
Running	Moderate+	25	3425	137
Greek Folk Dancing	Moderate+	40	2040	51
Walking	Low+	35	2492	71.2
Walking	Low+	25	875	35

**Table 2:** Sample activities provided by users and respective steps/min values

Following this initial categorisation and the expert’s guidelines, quality scores for each exercise were computed using two different Observation Quality (OQ) metrics:

<b>OQ_1</b>	<p>A Social Sensor observation corresponding to a steps/min value <math>s_{pm}</math> that is above a “truth” <i>ceiling</i> per activity category <math>c_{cat}</math> so that <math>s_{pm} \geq c</math> should be given the maximum accuracy score.</p> <p>Values below the <i>ceiling</i> should be given a gradually lower score</p>
<b>OQ_2</b>	<p>A Social Sensor observation corresponding to a steps/min value <math>s_{pm}</math> that is between a <i>min</i> and <i>max</i> value per activity category so that <math>min_{cat} \leq s_{pm} \leq max_{cat}</math> should be given the maximum accuracy score.</p> <p>Values such that <math>s_{pm} &lt; min</math> and <math>s_{pm} &gt; max</math> value should be given a gradually lower score</p>

In practice, OQ\_1 penalises only users who over-report their activities, while OQ\_2 penalises both under- and over-reporting. Both metrics are useful for different use cases and as such, were both included in the ESS pilot (for example, OQ\_2 can be

used to determine if users can understand the difference between activity intensities).

For both OQ metrics, the ceilings were set, in collaboration with the expert, at:

$$c_{Low} = 60 \text{ steps/min} \quad || \quad c_{Moderate} = 80 \text{ steps/min}$$

This was established by taking into account:

- relevant literature which agrees that approximately 100 steps/min can be treated as the equivalent of a 3 MET (moderate intensity) activity ([113]–[115])
- the fact that low intensity exercises such as walking lie in the 2+ MET Range [116].
- the user’s inexperience in providing self-reported activity data and
- the fact that exercise activities were not broken down into further sub-categories (a limitation which is further discussed in Appendix A).

For the above reasons, ceilings were slightly discounted from their literature-computed “standards” (e.g. 80 for 3+ MET instead of 100). Naturally, a “ceiling” concept introduces the problem of how to assess the quality of values that are below it. One way would be to assess them as 0% accurate but this is too penalising for self-reported data, which is expected to carry some amount of noise and error [117].

Instead, users’ exercises were assessed according to how close to the ceiling they are, by borrowing the concept of Fuzzy *membership functions* (mf) [118], which quantify the degree of membership of a value in a pre-defined set<sup>37</sup>. Given the lack of a training dataset, the mfs used in the ESS pilot were a result of following the “psychological” method of eliciting mfs [119] in collaboration with the expert.

## The OQ\_1 Metric

For OQ\_1, statements below the ceiling are given gradually lower membership degrees in the (0,1] range using a *Sigmoidal* membership function<sup>38</sup> (**Equation 1**), to make sure that values close to the ceiling are given a fairer score.

---

<sup>37</sup> This is by no means a full Fuzzy system approach – only the mf concept is reused to assign numerical values to textual concepts.

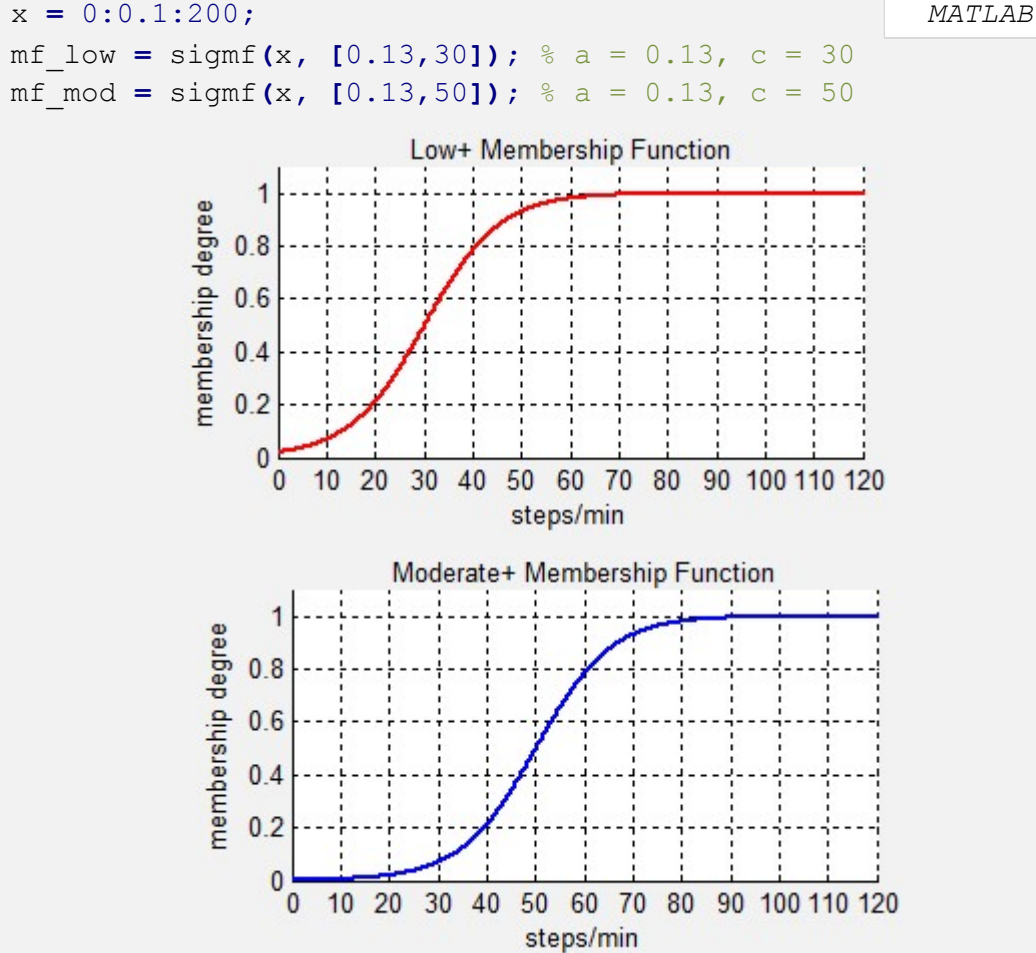
<sup>38</sup> <http://www.mathworks.com/help/fuzzy/sigmf.html>

$$f(x; a, c) = \frac{1}{1 + e^{-a(x-c)}}$$

(1)

Sigmoidal Function

**Figure 35** shows the resulting membership functions as plotted in MATLAB using the following parameters:



**Figure 35:** Membership functions for Low+ and Moderate+ Exercise Activities using the OQ\_1 Metric

For example, using the above Moderate+ mf:

- a statement of “Running” at 90 steps/min is **100%** accurate
- a statement of “Running” at 65.9 steps/min is **87%** accurate
- a statement of “Running” at 42.8 steps/min is **28%** accurate.

**Table 3** shows the result of applying the OQ\_1 metric to the activities of **Table 2**.

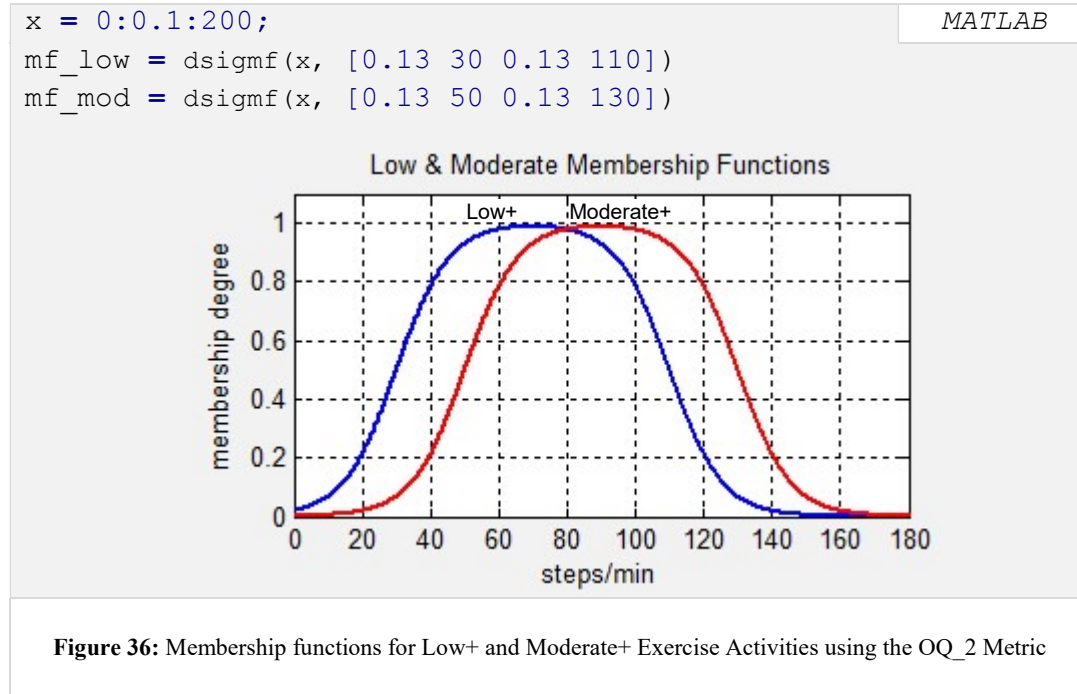
Name	Type	$s_{pm}$	OQ_1_score	Comment
Dancing	Moderate+	100.6	1.00	
Running	Moderate+	137	1.00	
Greek Folk Dancing	Moderate+	51	0.53	Over-reporting
Walking	Low+	71.2	1.00	
Walking	Low+	35	0.66	Over-reporting

**Table 3:** OQ\_1 scores of sample activities provided by the users

### The OQ\_2 Metric

For the OQ\_2 metric, statements that are both below and above the ceiling were given gradually lower membership degrees using the *difference between Sigmoidal functions*  $mf^{39}$ . This results in a more “traditional” fuzzy system approach (albeit with a sigmoidal curve instead of the classic trapezoid), where a value cannot concurrently be a 100% member of both the Low and Moderate activity fuzzy sets.

**Figure 36** shows the resulting membership functions as plotted in MATLAB:



<sup>39</sup> <http://www.mathworks.com/help/fuzzy/dsigmf.html>

As is expected, this results in lower quality scores for under-reporting, as can be seen in **Table 4**, which shows the result of applying the OQ\_2 metric to the activities of **Table 2**.

Name	Type	$s_{pm}$	OQ_2_score	Comment
Dancing	Moderate+	100.6	1.00	
Running	Moderate+	137	0.28	Under-reporting
Greek Folk Dancing	Moderate+	51	0.53	Over-reporting
Walking	Low+	71.2	1.00	
Walking	Low+	35	0.66	Over-reporting

**Table 4:** OQ\_2 scores of sample activities provided by the users

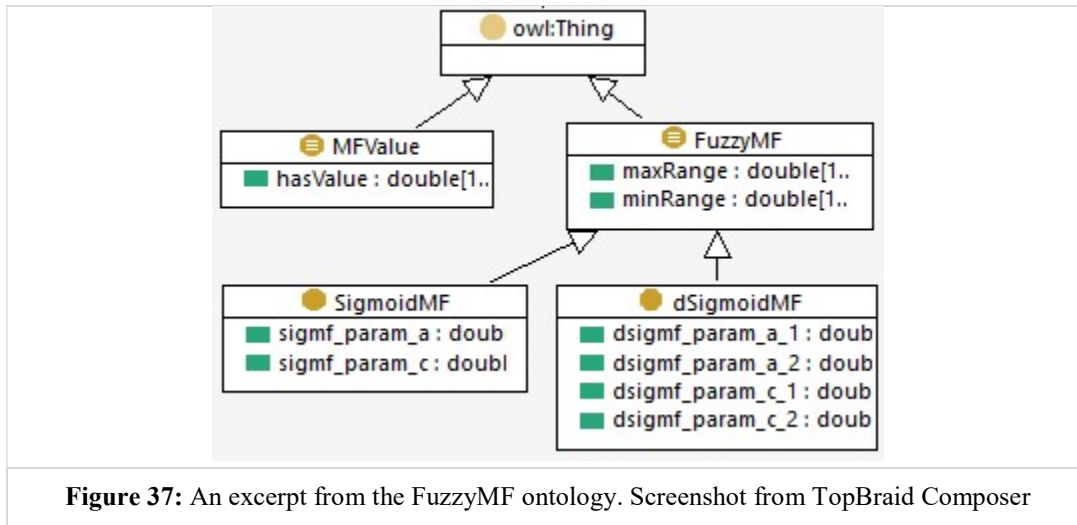
### 5.4.2 Using SOSENS to model Social Sensor Observation Quality

The previous chapter presented the methodology for determining social sensor observation quality that was followed in the NutriHeAl ESS pilot which, although developed specifically for the purposes of the pilot, can easily be applied to other social sensing scenarios. Similarly to the ESS observation data collected, SOSENS (with its extensions) can be used to enhance the social sensing process, by modelling the procedure, the data acquired and the related provenance chains in reusable RDF. This chapter discusses how SOSENS-T can be applied towards this goal.

As was the case with sensor observations, an initial bootstrapping process is required to describe the domain-specific characteristics of the pilot. To this effect, chapter 5.3 showed how this can be achieved using a variety of pilot-specific OWL constructs. However, *other ontologies* (usually implementation-agnostic) can also easily be (re-)used in SOSENS & SOSENS-T by simply assigning the correct properties (for example, an external rating methodology should be the target of `sosens-t:hasRatingMethod`). Social Sensor Observation Quality was expressed using the latter approach, as shown below.

## Bootstrapping

For describing the domain-specific characteristic of OQ\_1 & OQ\_2, a custom ontology, `FuzzyMF`<sup>40</sup>, was used. As can be seen in **Figure 37**, the `FuzzyMF` ontology describes the fuzzy `mf` concept *independently of the NutriHeAl ESS pilot*, with each type of `mf` being a subclass of the `FuzzyMF` OWL Class.



All `mf` subclasses, such as `SigmoidMf`, have:

- 1) *Common* OWL properties which are defined in the parent `FuzzyMF` class via OWL restrictions (such as `minRange` and `maxRange`) and
- 2) *Custom* OWL properties according to the specificities of each `mf` (such as `sigmf_param_a` and `sigmf_param_c` for the `[a,c]` parameters of the Sigmoidal function).

The ontology also includes an `MFValue` class (the degree of membership) that is restricted via `xsd:minInclusive/maxInclusive` to the `[0,1]` space.

Using the above, the `mfs` used for each metric were created as instances of the `SigmoidMF` & `dSigmoidMF` classes, an example of which can be seen below (for the OQ\_1 metric):

<sup>40</sup> <http://phd.pagkalos.com/sw/fuzzymf>

```

:OQ1_LowActivityMF a fuzzyMF:SigmoidMF;
  fuzzyMF:maxRange "200"^^xsd:double ;
  fuzzyMF:minRange "0"^^xsd:double ;
  fuzzyMF:sigmf_param_a "0.13"^^xsd:double ;
  fuzzyMF:sigmf_param_c "30"^^xsd:double ;
  rdfs:comment "A membership function to satisfy the
  requirements of OQ_1 Low Activity metric in the NutriHeAl
  ESS pilot"^^xsd:string ;
.

```

RDF/Turtle

```

:OQ1_ModerateActivityMF a fuzzyMF:SigmoidMF ;
  fuzzyMF:maxRange "200"^^xsd:double ;
  fuzzyMF:minRange "0"^^xsd:double ;
  fuzzyMF:sigmf_param_a "0.13"^^xsd:double ;
  fuzzyMF:sigmf_param_c "50"^^xsd:double ;
  rdfs:comment "A membership function to satisfy the
  requirements of OQ_1 Moderate Activity metric in the
  NutriHeAl ESS pilot"^^xsd:string ;
.

```

RDF/Turtle

## Modelling Social Sensor Observation Quality in SOSENS

This short bootstrapping procedure is the only requirement for using the SOSENS framework in the ESS pilot (as is the goal of a “semantic scaffolding” framework). **Figure 38** shows how simple it is for all of the parts of the ESS Social Sensor’s Observation Quality DQA process (which is, arguably, complex) to be described in rich RDF using SOSENS & its extension, SOSENS-T:

- 1 **Authority & Rating activity:** For each new `sioc:post`, a `TrustAuthority` is associated with a `RatingActivity`
- 2 **Rating activity provenance:** The `RatingActivity` `prov:used` all the `ssn:Observations` related to each `sioc:Post` (both from the Fitbit Zip & the Social Sensor), which establishes a provenance chain in regards to the data used to compute the rating
- 3 **Rating Methodology:** The `RatingActivity` followed a specific `RatingMethod` (e.g. `SigmoidMF`) in order to generate a `Rating`, which establishes a provenance chain between the rating and the specific method used.



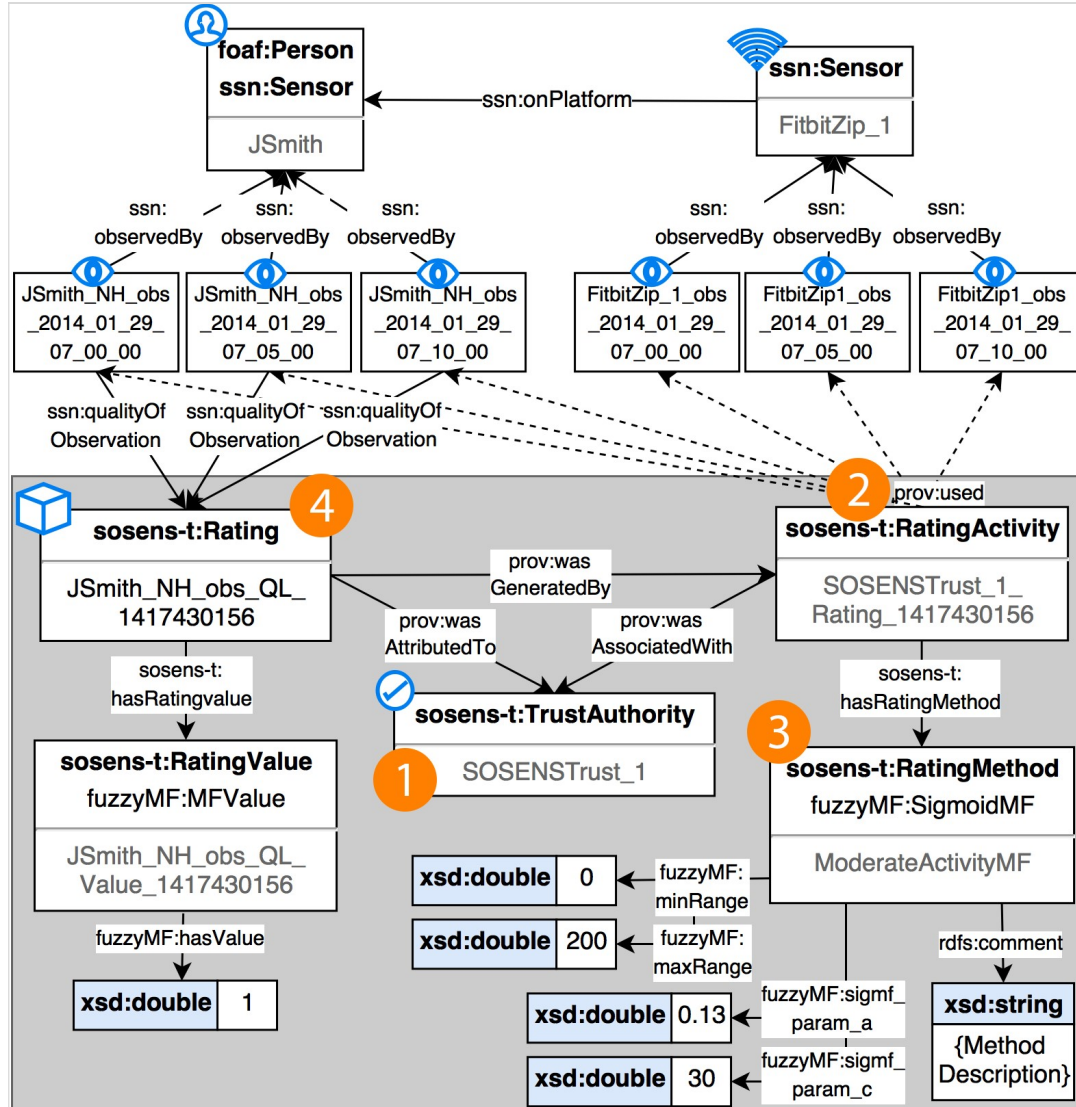


Figure 38: Using SOSENS-T to model Social Sensor Observation Quality in the ESS Pilot

**4 Rating & quality of Observation:** Finally, this Rating is assigned as the `ssn:qualityOfObservation` of all Social Sensor Observations that were parsed from the original `sioc:Post`

## Software Implementation

In order to implement the Observation Quality DQA process, a SOSENS-Trust Data Processor (DPr) SBB was deployed (see Chapter 4.2.3), in addition to the SOSENS SBBs described in the previous chapter (DPa, SWC, SDS). Given the above modelling and the observation modelling performed in the previous chapter, the following pseudo-algorithm was used to implement the OQ\_1 & OQ\_2 metrics:

<b>OQ_STEP_1</b>	For each sioc:Post by <user>, find the ssn:Observations that were parsed from it
<b>OQ_STEP_2</b>	For each ssn:Observation from OQ_STEP_1, find the <steps> value of the equivalent ground truth ssn:Observation, which is the observation that: <ul style="list-style-type: none"> <li>- is observed by a Fitbit_Zip</li> <li>- has &lt;user&gt; as the ssn:featureOfInterest</li> <li>- is observed at the same XSDDateTime</li> </ul>
<b>OQ_STEP_3</b>	Aggregate all values from OQ_STEP_2 and compute mean steps/min
<b>OQ_STEP_4</b>	Calculate accuracy score by comparing OQ_STEP_3 result with chosen RatingMethod according to exercise category & chosen Metric
<b>OQ_STEP_5</b>	Assign value of OQ_STEP_4 as the ssn:qualityOfObservation of each observation from OQ_STEP_1

Thus, for each new sioc:Post, a Python script (semantically described above as SOSENSTrust\_1, a prov:Software Agent) executed SPARQL queries such as the one shown in the code snippet below to retrieve all observations related to the post in question (**OQ\_STEP\_1**):

```
SELECT ?time ?value ?feature
WHERE {
  ?obs sosens:wasParsedFrom :NH_post_1;
  ssn:observationResultTime ?time;
  ssn:sensorOutput [ ssn:hasValue [ ?prop ?value]];
  ssn:featureOfInterest ?feature
}
```

SPARQL

The time & feature of the observation was then used to retrieve the value from the equivalent observation of a Fitbit Zip (**OQ\_STEP\_2**):

```
SELECT ?value
WHERE {
  ?obs ssn:wasObservedBy [a FitbitZip];
  ssn:observationResultTime <time from above query>;
  ssn:featureOfInterest <feature from above query>;
  ssn:sensorOutput [ ssn:hasValue [ ?prop ?value]];
}
```

SPARQL

The script aggregated the values, computed the mean (**OQ\_STEP\_3**) and then executed an external MATLAB script that returns the membership grade of the mean steps/min value to the equivalent membership function<sup>41</sup> (**OQ\_STEP\_4**).

Finally, using the SOSENS API, this membership grade was assigned as the quality of each observation (**OQ\_STEP\_5**).

### 5.4.3 DQA - Social Sensor Trust

#### Pilot Methodology

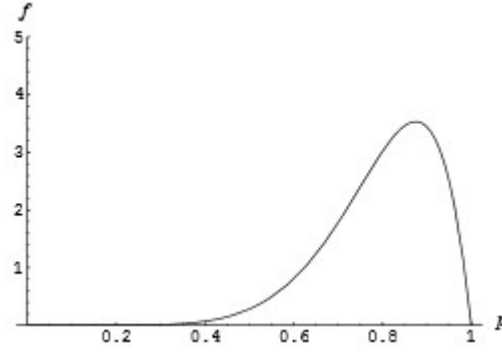
The previous chapter described the DQA process to obtain a quality rating and its implementation in SOSENS. With this value available for every social sensor's observation, it becomes possible to establish and describe a Trust aggregation methodology. As an end goal, the quality rating of each sensor's observations for a specific property should be aggregated over time to form its reputation as a trusted (= accurate) sensor for observing this specific property. Thus, in the NutriHeAl ESS pilot, the quality ratings of a participant's statements on Facebook about his/her activity should be aggregated over time to form its reputation as a trusted sensor for observing exercise. As discussed previously, there are many ways to aggregate these ratings into a score.

For the purposes of the pilot the primary approach was a Bayesian Reputation System (BRS) [120], [121] which can be used to predict future behaviour [122] and has been shown to be appropriate for modelling participant data collection habits [123]. One of the basic advantages of Bayesian systems is that they provide a *theoretically sound* basis (based on probability theory) for computing reputation scores.

“Classic” BRSs are *binomial* – they take binary ratings as input (i.e. positive or negative), and are based on computing reputation scores by statistical updating of *Beta* Probability Density Functions (PDF). The *a posteriori* score is computed by combining the *a priori* score with the new rating and can be represented in the form of the beta PDF parameter tuple  $(\alpha, \beta)$  or in the form of the probability expectation value of the beta PDF [120]. For example, a process with two possible outcomes  $(x, \bar{x})$  that has produced  $x$  7 times and  $\bar{x}$  1 time will have a beta function as shown in **Figure 39** and a probability expectation value  $E(p) = 0.8$ .

---

<sup>41</sup> This can be considered equivalent to using the `find` command for the value's index in the `x = 0:0.1:200` vector against the `sigmf` & `dsigmoid` functions shown before.



**Figure 39:** Beta function of event after 7 positive and 1 negative outcome.

A binomial BRS (bBRS), thus, works by collecting the amount of positive and negative feedback about an entity and uses **Equation 2** as an indication of how a user T is expected to behave in future transactions.

$$E = (r_T^X + 1) / (r_T^X + s_T^X + 2)$$

*where*

$r_T^X$  : the positive feedbacks from X about T

$s_T^X$  : the negative feedbacks from X about T

(2)

Probability expectation value of a user's reputation function (Binomial BRS)

In contrast to Binomial Bayesian Reputation Systems which allow ratings to be expressed with two values only (e.g. Good or Bad), *Multinomial* Bayesian Reputation Systems (*mBRS*) [124] use a *Dirichlet* PDF and allow ratings to be provided over  $k$  discrete levels (e.g. Bad – Mediocre – Average – Good – Excellent, or a 5-star rating approach).

$$\vec{S}(L_i) = \frac{\vec{r}(L_i) + W\vec{a}(L_i)}{W + \sum_{i=1}^k \vec{r}(L_i)}$$

*where*

$k$ : the number of discrete rating levels

$\Lambda = \{L_1, \dots, L_k\}$  : the set of  $k$  disjoint elements

$\vec{r}$ :  $k$ -component rating variable (the votes on the elements of  $\Lambda$ )

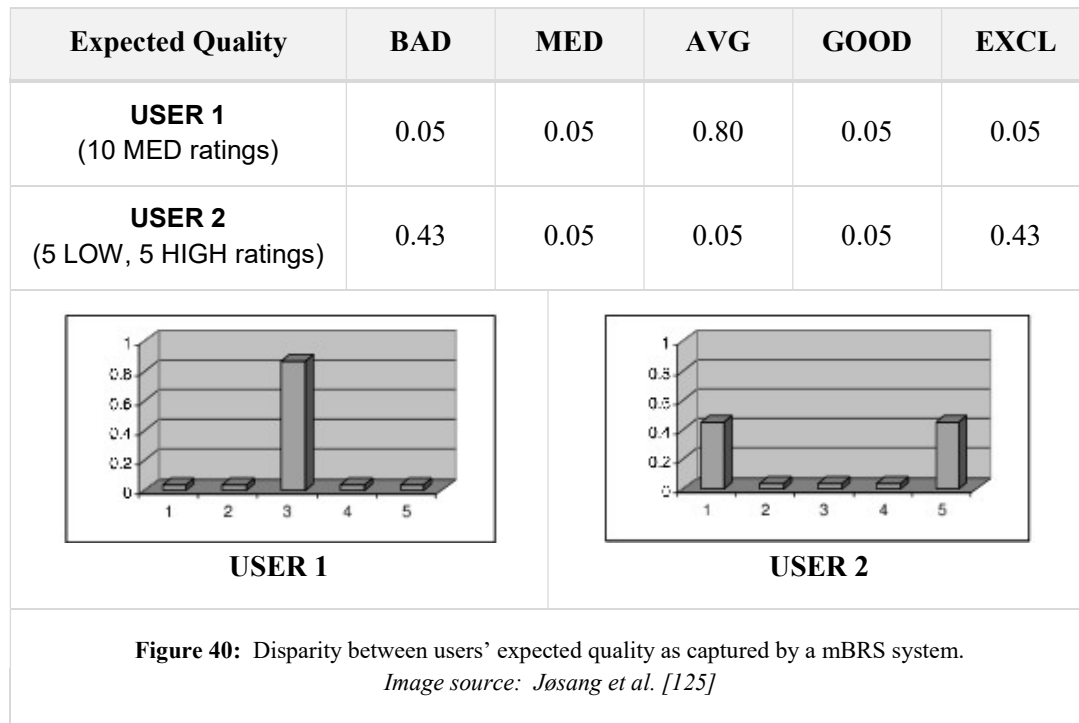
$\vec{a}$ : base rate vector

$W$ : the non-informative prior weight (typically set to 2)

(3)

Probability expectation value of a user's reputation function (multinomial BRS)

The resulting reputation is also expressed in the same multinomial way (as shown in **Equation 3**) which has the benefit of being able to distinguish between polarised and average ratings [121]. This means that users who constantly provide either really low-quality or really high-quality ratings can be differentiated from users who constantly provide average-quality ratings (e.g. see **Figure 40**). This is not possible in a bBRS and can prove useful in a later stage for classifying users and doing further analysis of their measuring capabilities and overall performance.



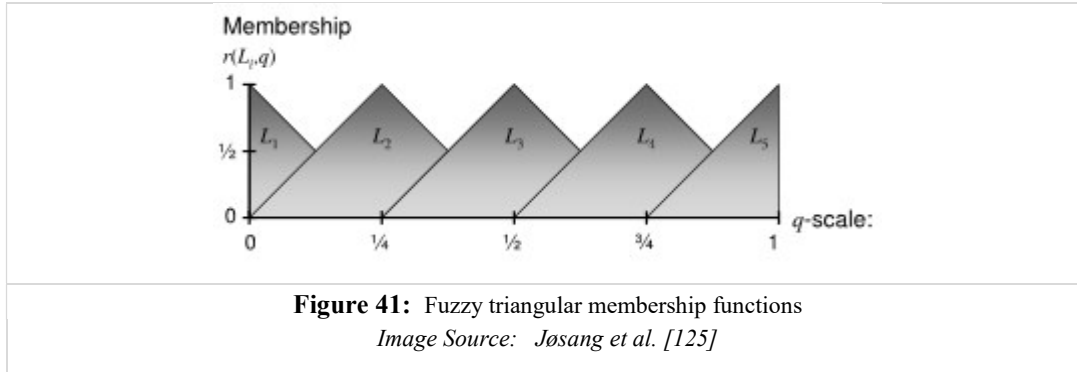
Values may also be presented as “point values” instead of the multinomial vector, for ease of understanding. In addition, mBRS systems:

- May be “fine-tuned” by modifying the base rate vector ( $\vec{a}$ ) as well as the non-informative prior weight ( $W$ ).
- Can support dynamic community base rates and/or individual base rates
- Can support aging (longevity factor) when aggregating ratings

### Customising a mBRS for the NutriHeAI ESS pilot

For all the reasons described above as well as its applicability to many other social sensing scenarios due to its implementation-agnostic nature, a mBRS system was used for determining Social Sensor Trust in the NutriHeAI ESS pilot.

Due to the fact that the quality of an observation in the ESS pilot is a *continuous value* rather than a value of multiple discrete levels, a fuzzy membership function methodology was applied, which is similar to the one used previously in this thesis as well as in one of the mBRS authors' follow-up paper [125]. This approach, also called a “sliding window”, allows a mBRS to accept continuous ratings as input by using triangular Fuzzy Membership functions for each rating level, as shown in **Figure 41**.



For the ESS pilot, five rating levels ( $L_1, L_2 \dots L_5$ ) were defined, with  $L_1$  representing “Bad” and  $L_5$  “Excellent” accuracy. Using the sliding window approach:

- An observation quality rating of 0.38 is entered into the mBRS system as the vector  $(0, 0.49, 0.41, 0, 0)$ , as a result of the membership degree in each of the triangular membership functions.
- After a few quality ratings have been entered, the resulting reputation score is computed by **Equation 3** and is described with another vector, which shows the probability expectation values of each element, e.g.  $(0.20, 0.00, 0.30, 0.00, 0.50)$

The above vector indicates that the Social Sensor’s observations have a 20% chance of being of “Bad” ( $L_1$ ) quality, a 30% of being “Average” and 50% of being “Excellent” ( $L_5$ ).

This vector can also be “converted” to a more human-readable format by assigning a point value to each rating level and computing the normalised weighted point estimate score (**Equation 4**) [125]. Although this leads to information loss, it allows the vector to be mapped to more traditional representations such as 1-5 stars or a probability.

$$\sigma = \sum_{i=1}^k v(L_i)S(L_i)$$

where

$$v(L_i) = \frac{i-1}{k-1}$$

(4)

Point estimate representation of a mBRS vector

In the NutriHeAl ESS pilot, resulting reputation scores were computed and stored both as the resulting mBRS vector and as a point estimate. The point estimate was calculated by assigning the following weights to each level (which results from **Equation 4** for  $k=5$ ):

BAD	MED	AVG	GOOD	EXCL
0.00	0.25	0.50	0.75	1.00

**Table 5:** Weights used for point value estimation in the ESS pilot

Using all of the information above, 3 different mBRS metrics were defined for the ESS pilot, according to the ratings provided as input:

<b>mBRS_All</b>	<p>A Social Sensor's trust value (i.e. the expected accuracy of all its observations) for determining <b>exercise</b> shall be defined as the result of a 5-level Bayesian Reputation System which sequentially receives <b>all of the user's exercise quality scores</b> as input.</p> <p>Multiple observations for the same event shall be treated as one unique score.</p> <p>Ratings shall be stored as a vector and a point value.</p>
<b>mBRS_Low</b>	As above but for <b>Low+ exercise</b> only.
<b>mBRS_Moderate</b>	As above but for <b>Moderate+ exercise</b> only.

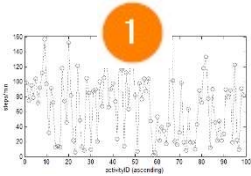
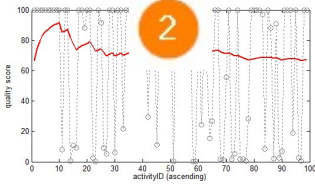


It is clear that each of the different mBRS metrics defined determines a different kind of *functional* trust in the capabilities of a social sensor. This was a research

requirement of the ESS pilot, in order to also be able to determine the capabilities of each user's exercise sensing capabilities separately. Further discussion of this aspect and the disparity between user sensing capabilities can be found in **Appendix A**. Another design decision was to only include one of each of the observation quality scores pertaining to the same `sloc:Post`, in order to determine accuracy at the exercise-level. Future studies may determine accuracy at the 5-minute level.

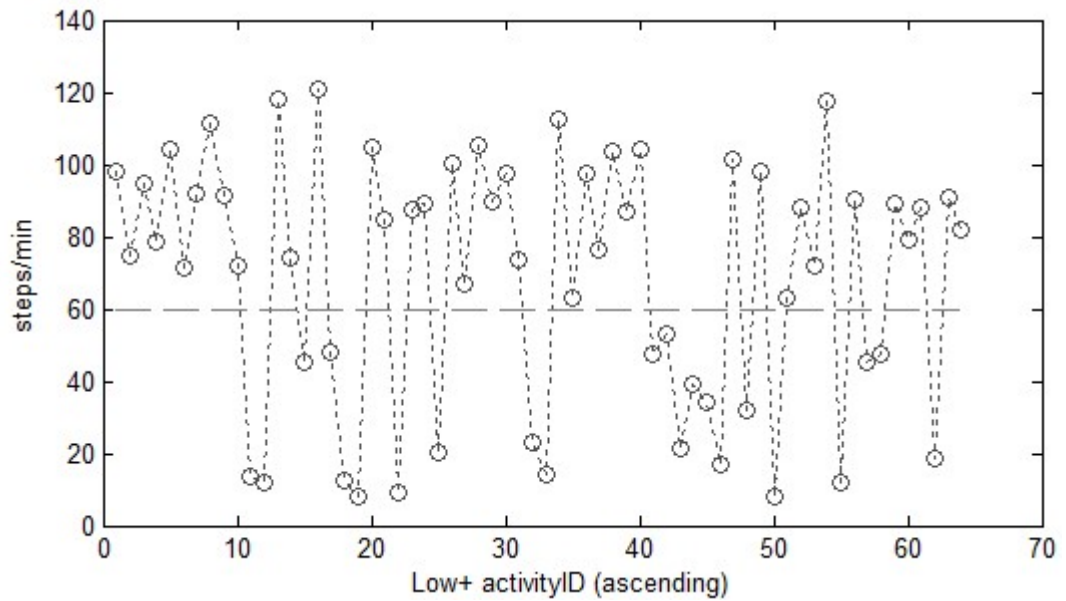
Finally, it should be noted that the methodology described & used in the ESS pilot is similar to ones found in e-Health systems such as Hedaquin [126] & MDRA (Medical Data Reliability Assessment) [127] systems, albeit with a common, central trust & reputation provider.

### An example

In order to better describe this part of the DQA process, the full-page information panels that follow show:

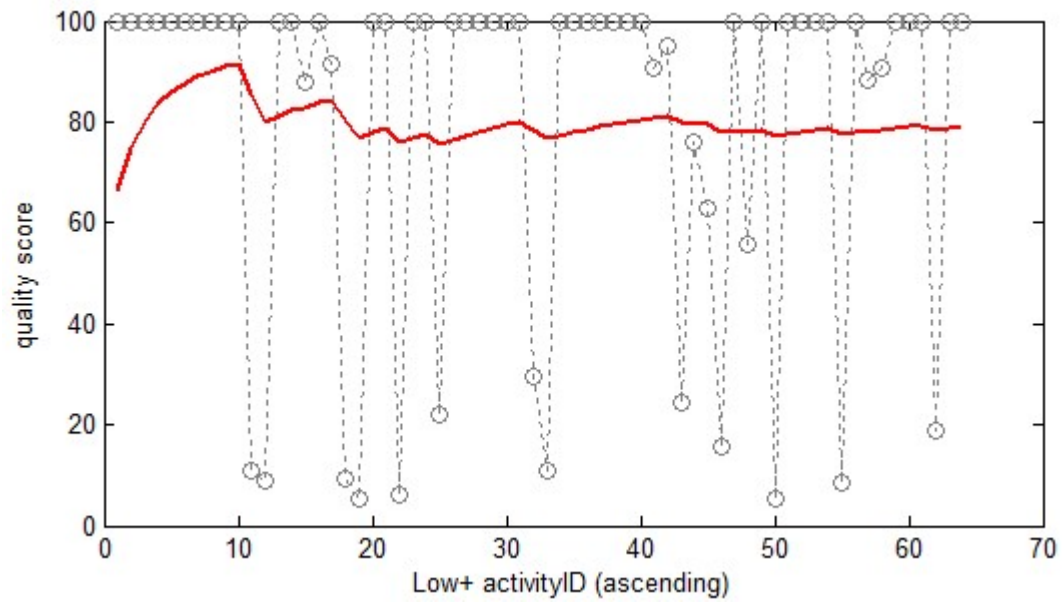
Panel	Description
	<p>A graph of anonymous user submitted exercise activities per category. Each exercise's value on the graph is the steps/min as reported by the Fitbit. According to the OQ1_metric, anything "above" the line constitutes a 100% accurate statement</p>
	<p>A Graph of the respective OQ1 quality scores for each of the above activities. The Red line shows the progress of the mBRS metric (normalised to 0-100)</p>
	<p>This table shows the final, multinomial Beta Reputation of the user for each level</p>
<p><b>Final point</b>  <b>e: 100%</b></p>	<p>Finally, the point estimate of the above vector is shown</p>



**DQA example: Low+ Activities (64)**

**Above:** Steps/min values for a user's Low+ Activities. OQ\_1 ceiling visible at 60 steps/min

**Below:** Computed Quality score using OQ\_1 & mBRS\_All



**Final mBRS vector:**

**BAD**

**MED**

**AVG**

**GOOD**

**EXCL**

0.10

0.12

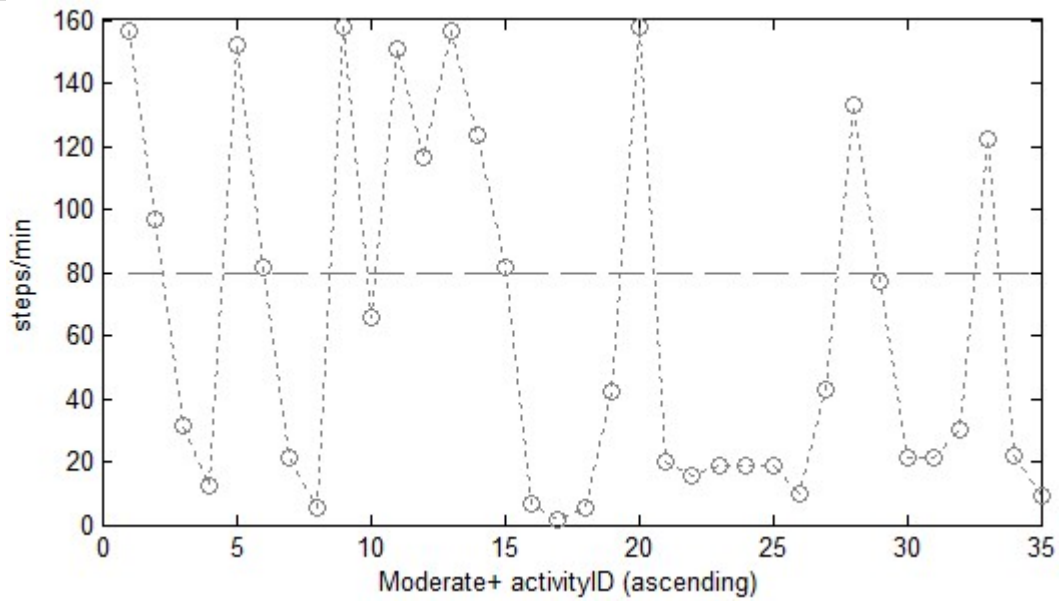
0.03

0.07

0.70

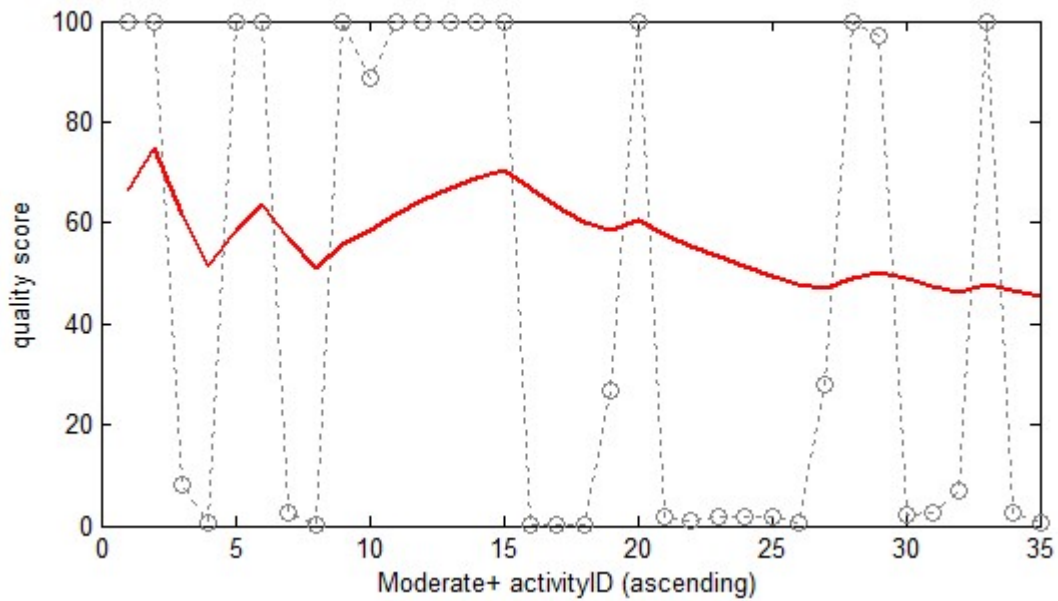
**Final point value:**

**79%**

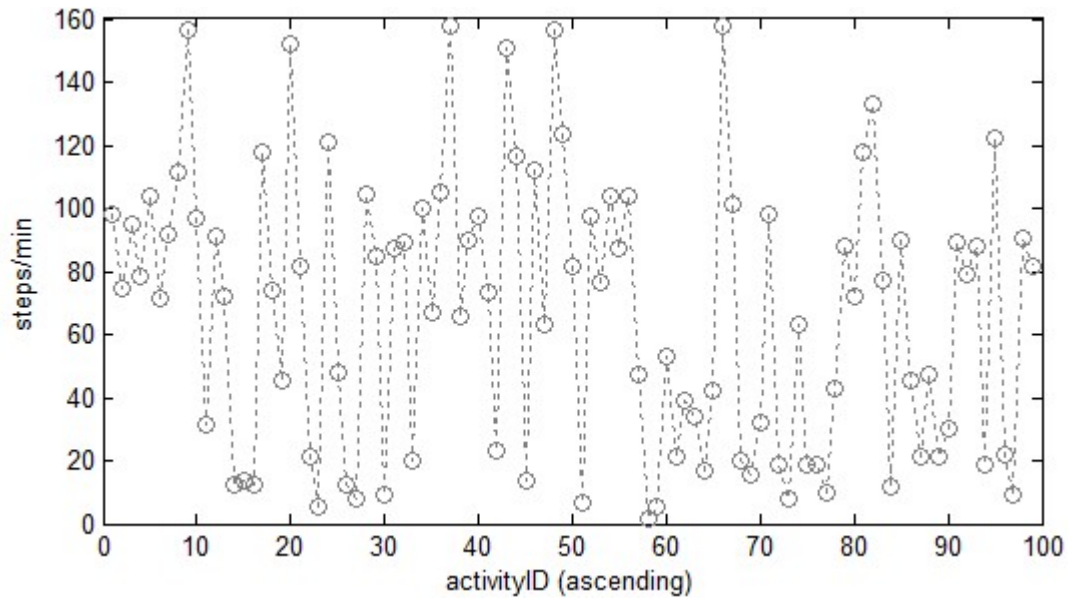
**DQA example: Moderate+ Activities (35)**

**Above:** Steps/min values for a user's Low+ Activities. OQ\_1 ceiling visible at 80 steps/min

**Below:** Computed Quality score using OQ\_1 & mBRS\_All

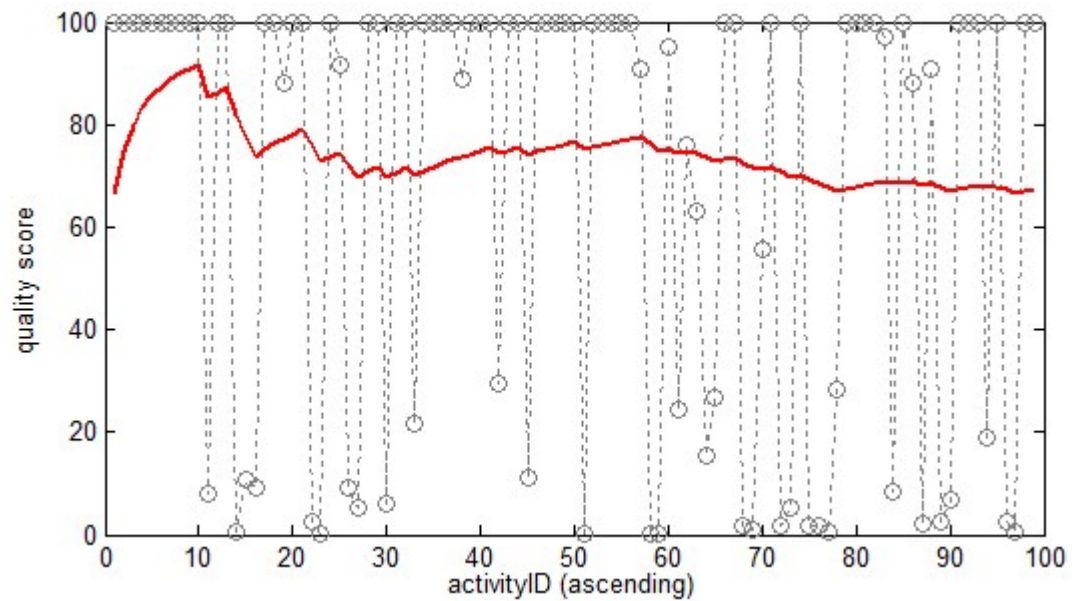


Final mBRS vector:	BAD	MED	AVG	GOOD	EXCL
	0.46	0.10	0.02	0.03	0.40
Final point value:	<b>45%</b>				

**DQA Example: All Activities (99)**

**Above:** Steps/min values for a user's Low+ Activities.

**Below:** Computed Quality score using OQ\_1 & mBRS\_All



**Final mBRS vector:**

**BAD**

**MED**

**AVG**

**GOOD**

**EXCL**

0.23

0.10

0.02

0.05

0.60

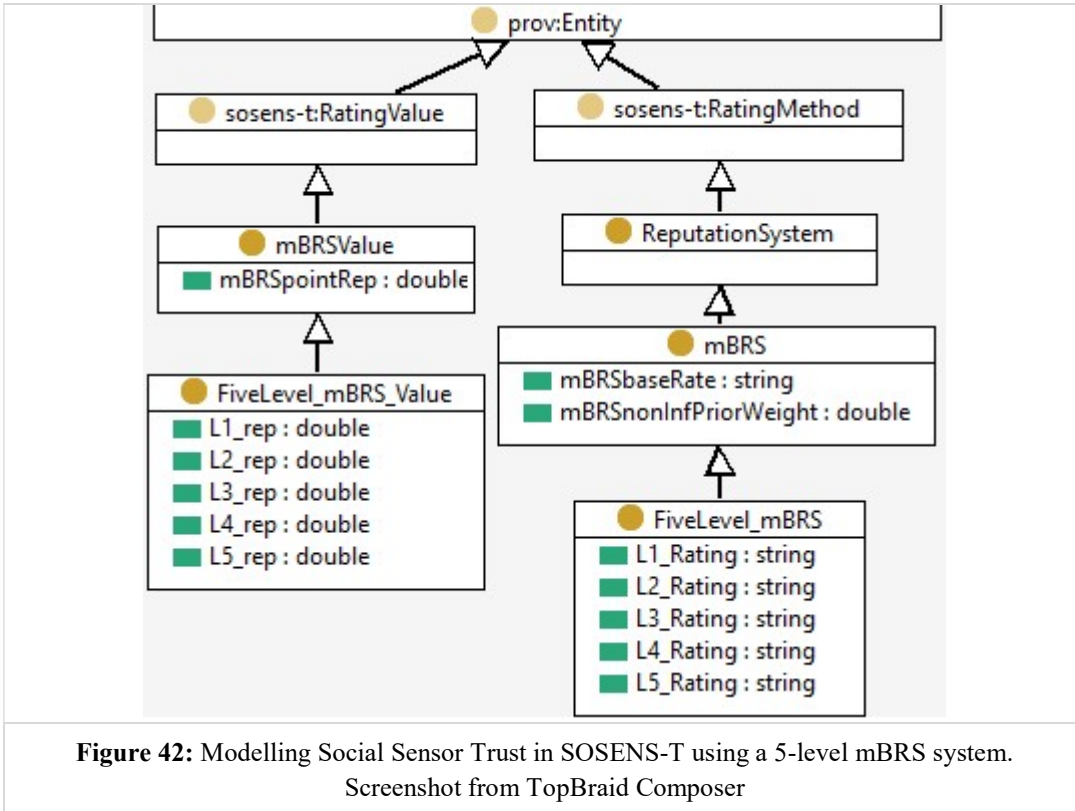
**Final point value:**

**67%**

#### 5.4.4 Using SOSENS to model Social Sensor Trust

Due to the abstract nature of SOSENS-T, it can be used to model both Social Sensor Observation Quality (as shown in chapter 5.4.2), as well as Social Sensor Trust. This section describes the implementation of the ESS pilot's DQA methodology using a mBRS system and SOSENS-T.

##### Bootstrapping



The domain-specific characteristics of using a 5-level mBRS system can be represented using the classes & properties shown in **Figure 42**. For the purposes of rapid bootstrapping, a `mBRS` is simply a `ReputationSystem` which is defined as a subclass of `sosens-t:RatingMethod`. Each mBRS system has some basic properties that characterise it such as `mBRSbaseRate` & `mBRSnonInfPriorWeight`. A value class, `mBRSValue`, was also defined, serving as the `rdfs:domain` for the `mBRSpointRep` property, which expresses the value as a point-estimate.

A specialisation of such a system (a subclass of `mBRS`) is the `FiveLevel_mBRS` which has five properties that describe the level of each rating (`L1_Rating`,

L2\_Rating etc.). Finally, a value for such systems is defined as a subclass of the generic mBRS value class and the domain for five properties (one for each level rating).

## Modelling Social Sensor Trust

Using the above, the following OWL instance was used to model the mBRS system used in the ESS Pilot:

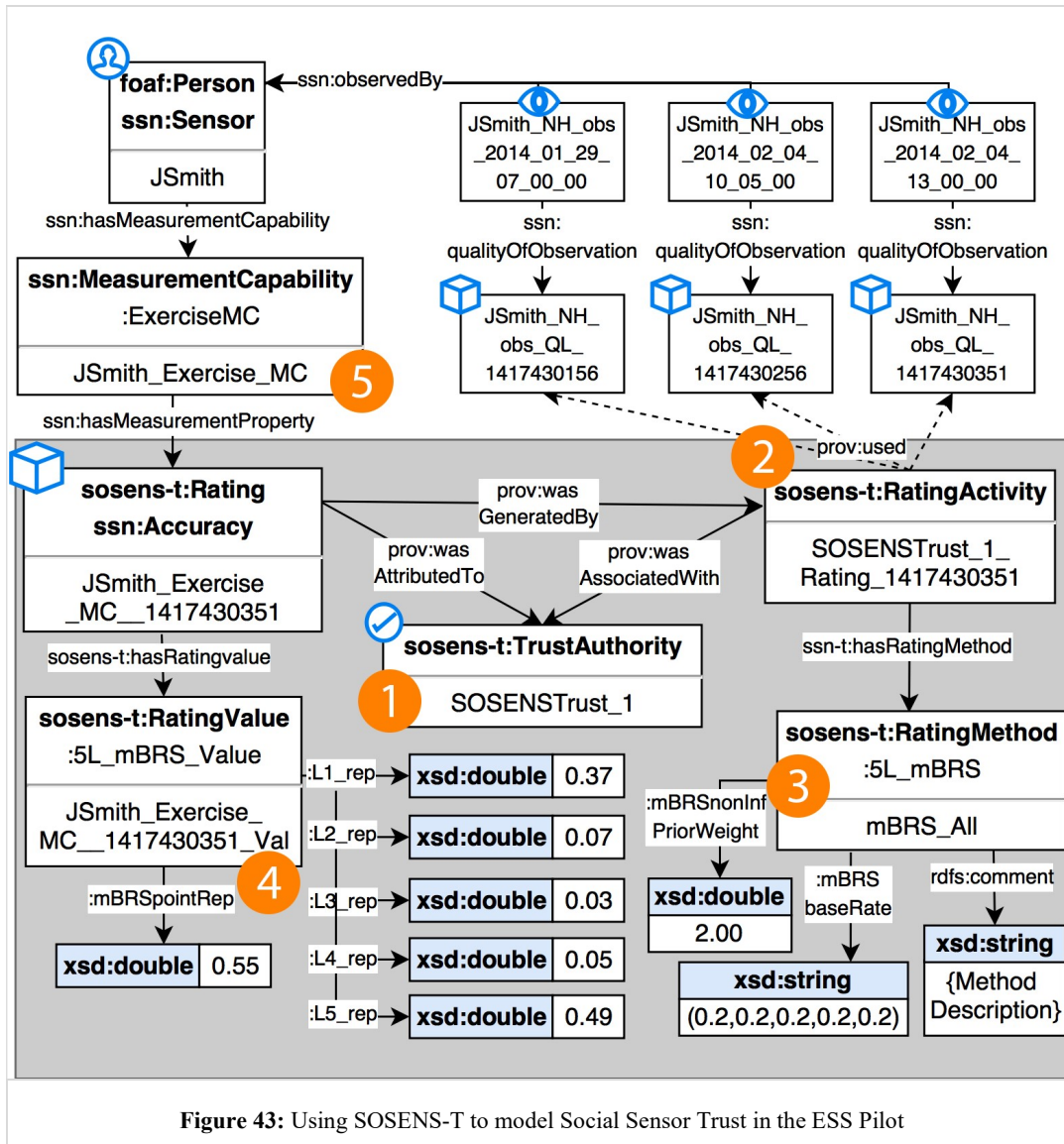
```
:NutriHeal_Exercise_mBRS a :FiveLevel_mBRS
:L1_Rating "Bad"^^xsd:string ;
:L2_Rating "Mediocre"^^xsd:string ;
:L3_Rating "Average"^^xsd:string ;
:L4_Rating "Good"^^xsd:string ;
:L5_Rating "Excellent"^^xsd:string ;
:mBRSbaseRate "(0.2,0.2,0.2,0.2,0.2)"^^xsd:string ;
:mBRSnonInfPriorWeight "2"^^xsd:double ;
rdfs:label "The 5-level mBRS used for the NutriHeal ESS
Pilot"^^xsd:string ;
.
```

Subsequently, **Figure 43** shows how SOSENS-T can be used to describe each part of the DQA process. The TrustAuthority - RatingActivity - Rating nexus is used in the same way as it was used in modelling Observation Quality. The main difference is that:

- The produced Rating is declared as the MeasurementCapability of a sensor (via hasMeasurementProperty) and
- The RatingActivity uses qualityOfObservation as input (instead of multiple observations)

According to the RatingMethod used, a different measurementCapability was created and/or updated. As a result, each user had an Exercise\_MC (for all exercises) and a Low\_Exercise\_MC & Moderate\_Exercise\_MC (for Low+ & Moderate+ activities respectively):

**1 Authority & Rating activity:** A TrustAuthority is associated with a RatingActivity for each new ssn:Observation by a Social Sensor, (one for each sioc:Post)



- 2 Rating activity provenance:** The `RatingActivity` `prov:used` all the `ssn:qualityOfObservation` of the Soc. Sensor's past observations
- 3 Rating Method:** The `RatingActivity` follows the `5L_mBRS` `RatingMethod` in order to generate a Rating
- 4 Rating Value:** The actual value of the rating is recorded both as a vector and as a point value, as described in the pilot's methodology
- 5 Rating & Measurement Capability (Accuracy):** This Rating is then assigned as the related Exercise measurementCapability and its value as the `ssn:Accuracy` of the Social Sensor.



## Software Implementation

For the software implementation of the mBRS DQA process of the ESS pilot, a Semantic Processing SBB similar to the one used in the software implementation of Observation Quality DQA was deployed. Every time the user posted a new exercise, a new rating was computed. The following pseudo-code was used to implement the mBRS\_All, mBRS\_Low and mBRS\_Moderate metrics:

For **each new sioc:Post**:

<b>SST_STEP_1</b>	Retrieve the qualityOfObservation of each of the a user's ssn:Observations up to now (one per sioc:Post) that match the chosen mBRS criterion (All, Low only or Moderate exercises only). Sort in ascending chronological order
<b>SST_STEP_2</b>	Convert each value from SST_STEP_1 to a vector denoting membership in a 5-level mBRS using the sliding window approach
<b>SST_STEP_3</b>	Calculate reputation score & point-estimate by sequentially feeding the vector from SST_STEP_3 to a 5-level mBRS
<b>SST_STEP_4</b>	Assign value of SST_STEP_3 as the repective (according to chosen mBRS criterion) ssn:measurementCapability of <user>

In order to implement the above, a second function of the SOSENSTrust\_1 SoftwareAgent was programmed (in Python). For each new sioc:Post the following SPARQL query implemented **SST\_STEP\_1**:

<pre> SELECT ?o ?q WHERE {   {     SELECT (MIN(?obs) as ?o)  WHERE {       ?post sioc:has_creator :Jsmith_Facebook .       ?obs sosens:wasParsedFrom ?post ;         a :ExerciseObservation .     } GROUP BY ?post   }   ?o ssn:qualityOfObservation [     sosens-t:hasRatingValue [?prop ?q ]]   FILTER (?q &gt;= 0) } </pre>	SPARQL
--	--------

The above query also serves as an excellent example of SPARQL's expressiveness, as it was able to encompass the entirety of the **SST\_STEP\_1**, which would be a loop-within-a-loop statement in standard programming techniques, in just a singular query.<sup>42</sup>

As for **SST\_STEP\_2** & **SST\_STEP\_3**, they were implemented simultaneously by a MATLAB script shown below (where `fuzzyAccuracy.fis` is the fuzzy logic toolbox<sup>43</sup> implementation of **Figure 41**'s sliding window):

<pre>function [repTable, averageTable, finalMF] = mBRS(data, c, BR, pointVal)     fuz = readfis('fuzzyAccuracy.fis');     repTable = [ ]; sumTable = [0;0;0;0;0];     for i=1:length(data)         [output, IRR, ORR, ARR]= evalfis(data(i),fuz);         sumTable = sumTable+IRR;         posTable = (sumTable + c*BR)./(c+sum(sumTable));         point = sum(posTable.*pointVal);         repTable(end+1) = point;     end     finalMF = posTable; end</pre>	MATLAB
---	--------

Finally, using the SOSENS API, the resulting rep score (in both its vector and point-estimate representation) was expressed in RDF and imported to the GraphDB database as the `ssn:MeasurementCapability` of the user (**SST\_STEP\_4**)

## 5.5 Benefits of SOSENS

The previous chapters presented the design, methodology and implementation of the ESS Pilot using the SOSENS Framework. As shown, all of the pilot-specific results (social data, social & hardware sensor observations, quality scores, social sensor trust scores), which are of great interest to the e-Health professionals were converted to semantically-rich RDF using the SOSENS framework and are readily available for consumption by machines & humans alike.

<sup>42</sup> This was possible thanks to the SPARQL 1.1 Subqueries feature <http://www.w3.org/TR/sparql11-query/#subqueries>). The same technique can also be applied to the queries of the Observation Quality software implementation.

<sup>43</sup> <http://www.mathworks.com/products/fuzzy-logic/>



In addition to the benefits of the framework presented in **Chapter 4**, the section that follows presents some further key benefits of using SOSENS. Due to the fact that this thesis mainly discusses the SOSENS framework, it does not focus on specific outcomes of the pilot in regards to lifestyle/physical activity patterns, which are further discussed in Appendix A.

### Seamless Integration of Social & Hardware Sensors

By modeling users as Social Sensors that monitor the `PhysicalActivity` property, the SOSENS framework provides an implementation- and application-agnostic integration between social and hardware sensors. SOSENS and the use of OWL/RDF constructs and SPARQL queries provide full customization, data transparency and the possibility for data reuse. Data can be retrieved via implementation-agnostic SPARQL queries such the one shown below and presented to the user according to application needs.

*“show me all observations of all sensors on a user for a specific timeframe”*

<pre> SELECT ?obs ?sensor ?time ?valueProp ?value WHERE {   ?obs ssn:featureOfInterest :JSmith;     ssn:observationResultTime [       time:inXSDDateTime ?time ];   ssn:observedBy ?sensor;   ssn:observedProperty ?obsProp;   ssn:sensorOutput [     ssn:hasValue [       ?valueProp ?value ]] . FILTER ( ?time &gt; "2014-05-28T14:20:00"^^xsd:dateTime &amp;&amp;   ?time &lt; "2014-05-29T14:31:00"^^xsd:dateTime &amp;&amp;   isLiteral(?value)) } </pre>	SPARQL
--	--------

### Adding trust to the mix

With the SOSENS-T extension, it becomes possible to query a social sensing dataset such as the one of the NutriHeAl ESS pilot in novel ways. Taking the various stages of DQA assessment into account, which created quality ratings and trust scores, SPARQL queries can very easily limit observations according to a specific threshold. For example, the SPARQL query shown below retrieves all observations with a quality score above a threshold (e.g 0.75):

*“show me all observations that are of high quality”*

```
SELECT DISTINCT ?obs ?q
WHERE {
  ?obs a ssn:Observation;
    ssn:observationResultTime ?time;
    ssn:observedBy ?sensor;
    ssn:sensorOutput [ ssn:hasValue [
      ?valueProp ?value ]] ;
    ssn:qualityOfObservation [
      sosens-t:hasRatingValue [?prop ?q ]].
  FILTER (?q >= 0.75)
}
```

SPARQL

An ever more interesting approach would be to query for observations from trusted sensors only. For example, suppose that the users of the NutriHeAl pilot take part in another Social Sensing scenario, where *there is no DQA methodology*, thus observations are not accompanied by quality scores. In that case, their *reputation* can be used to filter observations from the data consumer side.

*“show me exercise observations from trusted sensors only”*

```
SELECT DISTINCT ?obs ?rep
WHERE {
  ?obs a ssn:Observation;
    ssn:observationResultTime ?time;
    ssn:observedBy ?sensor;
    ssn:sensorOutput [
      ssn:hasValue [
        ?valueProp ?value
      ] ] .
  ?sensor ssn:hasMeasurementCapability [
    a :ExerciseMeasurementCapability;
    ssn:hasMeasurementProperty [
      a ssn:Accuracy;
      sosens-t:hasRatingValue ?rep
    ] ] .
  FILTER (?rep >= 0.75)
}
```

SPARQL

## Semantic Database Linking

The semantic representation of data in SOSENS not only makes integration between local data available, but also allows for integrating data from other semantic databases. This can be done via SPARQL Federated Queries or by downloading the data and merging them with the local SOSENS instance if data is not time-sensitive.

For example, in the case of the NutriHeAl pilot, the original dataset can be enhanced by including obesity statistics per age group and gender from data acquired by the Hellenic Medical Association for Obesity (HMAO)<sup>44</sup> [128]. This allows for queries that can return both actual and statistical data, such as the Query shown below which returns the BMI (Body Mass Index) as measured by the NutriHeAl SNAApp and the statistical BMI (+ standard deviation) relative to Age & Gender from the HMAO database<sup>13</sup>:

*“show me the statistical BMI from HMAO for users of a specific Age & Gender”*

```
SELECT ?user ?age ?gender ?BMI ?statBMI ?SD
WHERE {
  ?user :Age ?age;
    :Gender ?gender;
    :BMI ?BMI .
  OPTIONAL {
    ?eiepStat eiep:Age ?age;
    eiep:Gender ?gender;
    eiep:StatBMI ?statBMI; #Statistical BMI
    eiep:SD ?SD . #Standard Deviation
  }}

```

SPARQL

Query result from TopBraid Composer ME:

[user]	age	gender	BMI	statBMI	SD
◆ foaf:NH_1	F 24	S M	F 25.3	F 24.7	F 4.7
◆ foaf:NH_2	F 40	S M	F 25	F 27.6	F 4.6
◆ foaf:NH_3	F 25	S F	F 21.8	F 22.6	F 4.4
◆ foaf:NH_4	F 43	S F	F 23.1	F 26.2	F 5.1

<sup>44</sup> Database not publically available so data was merged locally. For remote data, SPARQL Federated Queries should be used.

The possibilities of linking with external databases are abundant. For example:

- A semantic database with *weather* information can augment the data per day in order to see if there are correlations between activities performed and weather conditions
- A semantic database with GPS data can be queried to correlate *exercise* with *location*
- A semantic database with data from other Social Media can be used to compare similar posts
- Data from other experiments based on SOSENS (or similar platforms that export semantic data) can be easily integrated.

In regards to the latter, even though SW technologies & the use of SOSENS, as shown in this thesis, can aid in overcoming the heterogeneity which is prevalent in Social Sensing environments, there is always some degree of difference between implementation, based on the different decisions made when bootstrapping. However, OWL Inference mechanisms such as `owl:equivalentClass` (& `Property`), and `owl:sameAs` [49] can be used to bypass these by linking ontologies at the semantic level (sometimes called a “semantic join” [90]).

For example, suppose a query that retrieves Fitbit Zip data by looking for data produced by an instance of the `PhysicalActivitySensor` class:

<pre>SELECT ?obs WHERE {   ?obs observedBy [ a :PhysicalActivitySensor ] . }</pre>	SPARQL
--	--------

Now suppose that a different semantic database also includes results from Digital Pedometers that observe `:PhysicalActivity` but does not use the same Class name as a SOSENS instance, making queries such as the one above null. By defining the `PhysicalActivitySensor` as an *Inferred* OWL Class as was described in chapter 5.3 (and shown in the code snippet below), a sensor that `ssn:observes :PhysicalActivity` is automatically inferred to be a Digital Pedometer, making the query interoperable.

<pre><b>:PhysicalActivitySensor</b> a owl:Class ;   rdfs:subClassOf ssn:SensingDevice ;   owl:equivalentClass [     a owl:Restriction ;     owl:hasValue <b>:PhysicalActivity</b> ;     owl:onProperty ssn:observes ; ]</pre>	RDF/Turtle
---	------------

Because it is unrealistic to assume that everybody will use the exact same instance name to refer to the activity performed by a user (“that would require some grand design, which is contrary to the spirit of the web” [49]), `owl:sameAs` can be used to semantically link this instance to another database’s instance, such as `example:Steps`). This mechanism can also be used to link an object to more detailed information in other databases such as, for example, linking a local instance of `:BloodPressure` to the DBPedia entry `:Blood_Presure`<sup>45</sup>.

```
:BloodPressure owl:sameAs dbp:Blood_Presure.
```

*RDF/Turtle*

## Utilization of Social Data

With the expressive capabilities of SPARQL, a SOSENS data consumer can query the system for simple data retrieval or even traverse a social graph to get results from socially-connected participants. For example, one of the previous queries could easily be expanded into the one shown below which returns the physical activity data of both John Smith and all his social network connections

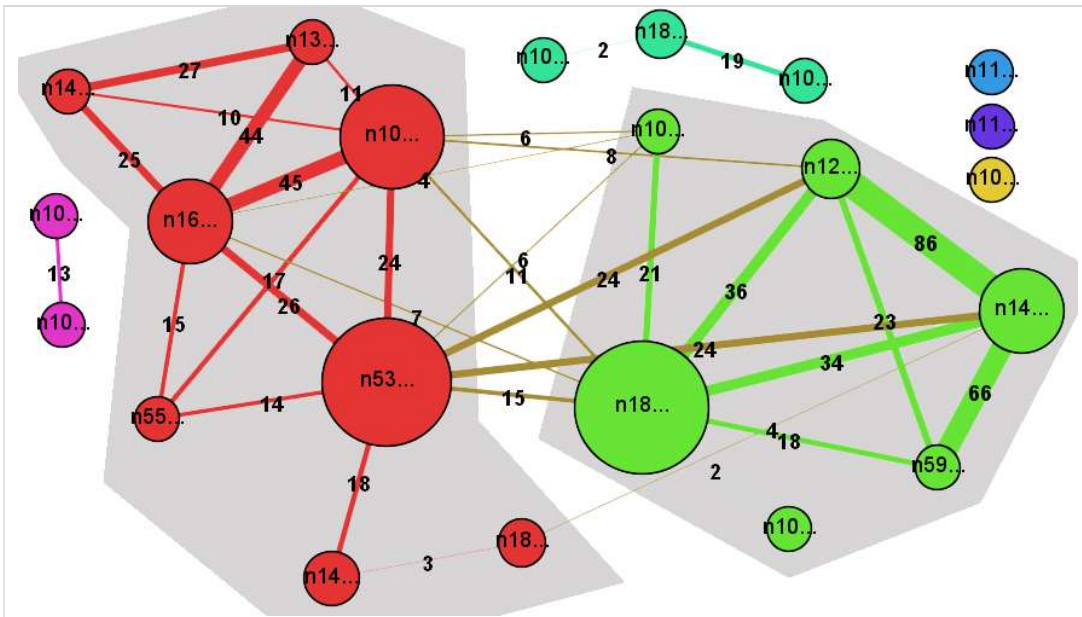
*“show me the physical activity observations of a user and his friends”*

```
SELECT ?obs
WHERE {
  ?user foaf:name "John Smith";
        foaf:knows ?friend .
  {
    ?obs ssn:featureOfInterest ?user;
        ssn:observedProperty :PhysicalActivity .
  }
  UNION
  {
    ?obs ssn:featureOfInterest ?friend;
        ssn:observedProperty :PhysicalActivity .
  }
}
```

*SPARQL*

This social data can always be further analysed using classic Social Network Analysis (SNA) techniques, as shown in **Figure 44**, which shows the results of a

<sup>45</sup> [http://dbpedia.org/resource/Blood\\_pressure](http://dbpedia.org/resource/Blood_pressure)



**Figure 44:** Social Network Analysis of the NutriHeAl Pilot, Round 1, using Gephi 0.8.212

minor analysis using three different SNA metrics that were applied to a subset of NutriHeAl pilot participants:

**Community Detection** (shown by coloured groups): As a result of determining the network's modularity (the strength of division of a network into groups) two large communities were detected in the pilot. This indicates a densely connected group and these communities can be used, for example, to further understand how users may affect one another when being monitored by a Dietician, or when wearing a wearable sensor

**Tie Strength** (shown by size & label of connecting edge): The strength of a connection between two users in this instance was calculated by the amount of mutual friends between them. This can result in a better understanding of the person-to-person connections in the pilot.

**Influential persons in Network** (shown by size of a node): Closeness Centrality (the measure of how long it will take to spread information from one node to all other nodes sequentially) can be used to understand a user's position in the network. If, for example, SOSENS is used in an intervention scenario, the network can be treated as an information flow network and this metric makes it possible to select the proper intervention target(s), especially in the case of a limited budget where intervention can't take place for all participants.

The above SNA techniques is just a subset of the possibilities in a dataset where social data is available and are presented here to display the possibilities of a SOSENS instance. The seminal work in [130] as well as recent research efforts in e-Health (e.g. [131],[132]) present even more opportunities for utilising social data.

## Chapter conclusions

This chapter discussed how a full-scale Exercise Social Sensing (ESS) pilot can be enhanced and augmented by SOSENS. The NutriHeAl ESS pilot was built around a custom Facebook app for data collection and novel Data Quality Assessment methodologies (both specific to this thesis), all of which were represented in SOSENS, with the appropriate provenance chains and attribution of actors and processes.

**Figure 45** shows the results of running the pilot implementation for an average of 35 days. The obtained exercise activities and related DQA values of the participants are very important for the specific application area to determine whether users can be trusted with self-reporting their activity.

Results (SOSENS – ESS instance)
<p>44 Participants over 35 days provided:</p> <ul style="list-style-type: none"> <li>- 44 foaf:Person and 19701 foaf:knows</li> <li>- 88 ssn:Sensor <ul style="list-style-type: none"> <li>o 44 Social Sensors</li> <li>o 44 FitbitZip</li> </ul> </li> <li>- 425,085 ssn:Observation <ul style="list-style-type: none"> <li>o 407,808 via FitbitZip</li> <li>o 17,277 via NutriHeAl from 6548 sioc:Posts</li> </ul> </li> <li>- 1654 sosens-t:Rating(mBRS) for Exercise Measuring Capability (MC) <ul style="list-style-type: none"> <li>o 569 sosens-t:Rating for Moderate+ Activity MC</li> <li>o 995 sosens-t:Rating for Low+ Activity Quality</li> </ul> </li> </ul>
<b>Figure 45:</b> Synopsis of the data gathered from the ESS pilot using SOSENS

One of the most important benefits of this work, however, is that all data collected from the Social Sensing experiment is not an “information silo”, such as the ones typically found around the Web. Using Semantic Web technologies, one can very

easily expose parts of the dataset using a SPARQL endpoint and allow researchers not only to access the data but also assess and review it, as it is accompanied by SOSENS metadata which can help decide if they are fit-for-use. As discussed in the “Advantages of SOSENS” chapter (5.5), this dataset can be *further* enhanced in a multitude of ways, such as linking to other databases and utilising the Social & Trust data to further analyse user-generated content.



## 6 CONCLUSIONS

This chapter summarises the research presented in this thesis and its contribution to the emerging field of Social Sensing.

The main focus of this thesis was using Semantic Web (SW) Technologies to model Social Sensing environments, effectively bringing user statements on Social Media to the same semantic level as “classic” hardware sensor observations and allowing the integration of the two, when needed.

To achieve this multi-faceted goal and tackle (i) the evident heterogeneity problem and (ii) the lack of provenance in most existing solutions, this thesis proposed the SOSENS family of high-level ontologies. Based on well-established SW methodologies, SOSENS describes *users as sensors* and their *statements as sensor observations*, with explicit SW objects & properties covering the *provenance* of each element. With the help of extensions such as the SOSENS-Trust ontology, it also paves the way for describing other critical processes within the same environment, such as Data Quality Assessment.

With the SOSENS family of ontology as a basis, the **SOSENS Framework** was introduced: a holistic, systemic and implementation-agnostic *software architecture* for modelling and managing Social Sensing environments. Using the framework and conforming to its requirements acts as a **semantic “scaffold”** for heterogeneous sensing environments and allows for the creation of novel, *interoperable* sensing spaces, as well as *re-usable* information.

The benefits of the framework were clearly presented and evaluated in a pilot implementation, part of the NutriHeAl project, where self-reported physical activity from Facebook Social Sensors was combined with Fitbit Digital Pedometers in order to track physical activity as well as calculate data quality and user trust.

Using the SOSENS framework, the semantically-rich data in complex Social Sensing spaces is efficiently collected, integrated, analysed and stored along with clear provenance & attribution metadata of all the actors & processes involved. This novel approach is a “by researchers - for researchers” framework which puts Social Sensing into the Web 3.0 Space and promotes data re-use.

## 6.1 Contributions

### **The SOSENS ontology: An implementation-agnostic model for Social Sensing**

There has been no holistic, implementation-agnostic approach to Social Sensing, resulting in troves of useful data from Social Sensing experiments, expressed in application-specific semantics and “trapped” within information silos. Using the SOSENS family of ontologies, researchers around the world can model and share their Social Sensing data in a uniform, machine-understandable way, bundled with valuable metadata that raise information quality.

Following the principles of the SW, the ontology is based on existing, well-established ontologies which facilitate its use, both in new as well as existing SW environments.

### **The SOSENS Framework: An abstract framework for Social Sensing**

The SOSENS framework showcases the capabilities of a SW approach and its applicability to Social Sensing. The framework is designed using Architecture Building Blocks, providing implementation-agnostic SW requirements. A reference implementation is, however, also presented within, for quick bootstrapping of a Social Sensing Space.

The SOSENS Web API, part of the framework, is a powerful, user-friendly API which allows even non-expert users to rapidly create SW data. Although designed for creating Social Sensing data, the same API can be used to create “traditional” sensing data as well, something which was missing from the Semantic Sensor Network community.

### **Complete Semantic Web approach**

The modelling & implementation issues are tackled entirely within the SW technology stack, which, to the best of the author’s knowledge, presents the first attempt of its sort, at least in this scale. Although many technologies that allow substituting parts of the SW stack exist, doing so takes away from some of SW solutions’ best benefits, such as the fact that they are storage-agnostic and a natural fit for abstract, open-ended problems.

It should be noted that the flexibility of SW applications comes with potential drawbacks which are well-known and discussed within the scientific community. Finding equilibrium between flexibility and performance/efficiency is probably one

of the most common problems in Computer Science and, undeniably in real life as well: if you want flexibility in your booking dates for an airplane ticket, you do end up paying a premium! As such, there can be cases where a problem with requirements of massive data scaling or rapid update transaction times can be more efficiently solved using more traditional technologies.

However, in such cases, there's nothing stopping a data producer from *also* describing his data using OWL and wrapping an existing database with SPARQL endpoints in order to integrate it in a broader SW environment. It may be some time before SW technologies become the norm, but there's realistically no reason to not take advantage of its many benefits right now, especially with the increasing capabilities and performance of Semantic Reasoners and Triples databases<sup>46</sup>.

### **A full-scale pilot implementation to support the approach**

The NutriHeAI pilot portrayed the many benefits of using the SOSENS framework, as presented in **Chapter 5.5** (seamless integration of social & hardware sensors, trust-enabled queries, semantic database linking, utilisation of social data etc.). These benefits are applicable not only to the e-Health domain, which was the pilot's focus, but to all Social Sensing scenarios because they are *inherent benefits* of the *SW technologies* behind the framework.

The pilot itself can also serve as a “how-to” example for prospective SOSENS framework users. Incidentally, the pilot was also one the most successful self-reported activity data-gathering initiatives of its scale in Greece and has created a dataset which is now actively studied by under- and post-graduate students of the Alexander Technological Educational Institute of Thessaloniki, department of Nutrition & Dietetics, in order to discover patterns of association between the social connections, lifestyle and physical activity of users.

## **6.2 Research Publications**

The research described in this thesis appears in the following peer-reviewed journals and conferences:

---

<sup>46</sup> An excellent showcase of current SW capabilities can be found in Oracle's Nov 2016 whitepaper regarding the “1 Trillion RDF triples” benchmark, where performance indexes such as  $1.527 \times 10^6$  Triples Inferred and Indexed / Second and  $1.130 \times 10^6$  Query Results / Second are reported: <http://bit.ly/1198flk> (Online PDF)

### Peer-reviewed Journals:

- I. Pagkalos and L. Petrou, “SENHANCE: A Semantic Web framework for integrating social and hardware sensors in e-Health,” *Health Informatics Journal*, vol. 22, no. 3, pp. 505–522, Sep. 2016<sup>47</sup>.
- I. Pagkalos and L. Petrou, “Scaffolding Social Sensing Environments Using the SOSENS Semantic Web Framework”, *International Journal on Semantic Web and Information Systems* (accepted with revisions)
- I. Pagkalos, A. Kokkinopoulou, M. Weal, L. Petrou, M. Hassapidou, "Exercise monitoring of young adults using a Facebook application", *Digital Health Journal* (accepted with revisions)

### Peer-reviewed international conferences:

- I. Pagkalos and L. Petrou, “Using Social Network Apps as Social Sensors for Health Monitoring,” in *XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013*, pp. 1330–1333.
- I. Pagkalos and M. Hassapidou, “Using social applications to complement physical activity monitoring,” *Obes. Facts*, vol. 6, no. s1, p. 31, 2013. (also in: *proceedings of the 20th European Congress on Obesity (ECO 2013)* 12–14 May, 2013, Liverpool, UK)
- I. Pagkalos, D. Rossiou, and S. Papadopoulou, “Monitoring physical activity through social networks: a Facebook case study,” *presented at the 6th DIETS/EFAD Conference*, Portoroz, Slovenia, 2012.

## 6.3 Future Work

### Rules & Inferencing

As showcased in the pilot, SW technologies were used for providing meaningful descriptions to the data collected (either directly, or via inference) and the bulk of the work for determining data quality, user trust and any other type of new, generated information was “outsourced” to MATLAB or external tools such as Python scripts. This was a conscious choice, given the currently “muddled” field of rule languages on the Semantic Web: Although RIF (Rules Interchange Format

---

<sup>47</sup> The SENHANCE framework was a previous title for the framework described within this thesis

[133]) is a W3C standard, it is (as its name gives away), focused on rule *exchange* rather than a one-fits-all rule language. Other approaches such as SWRL [134] and SPIN [135] are more promising but are still in the submission stage, and were thus, not included in the SOSENS framework.

However, given the increasing number of discussions within the scientific community and the W3C working groups, it is safe to assume that one language will emerge as a standard. A worthwhile future extension to SOSENS will be using rules to calculate values such as user trust *directly* via semantic reasoning which would make it more useful as a stand-alone solution. Some researchers (e.g. [136] and [137]) have already attempted something similar with SPIN, albeit with non-recursive data quality rules. One can envision that enhancing the framework with pre-built rules such as Fuzzy MFs and mBRS would increase its applicability and reach.

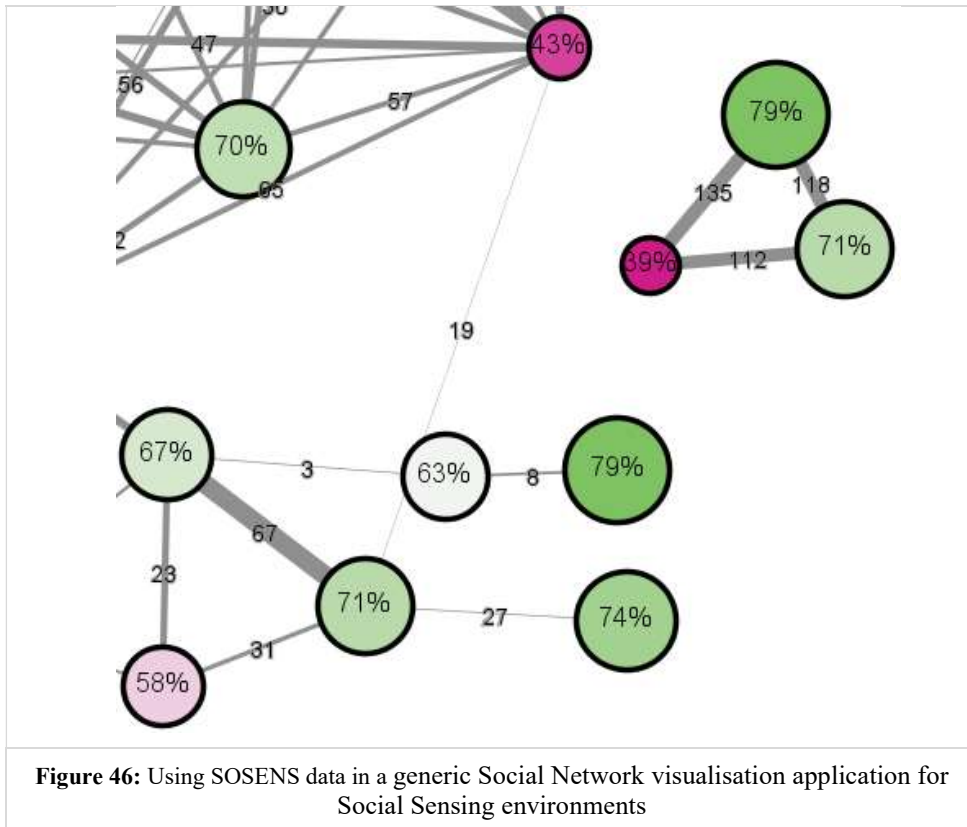
## Privacy

In the same spirit as above, Rules can be put in place in order to preserve user privacy. In most of the Social Sensing experiments, information about users is considered public, and if not, typical “privacy-preserving” measures are used, such as obscuring user information. However, Social Sensing spaces are very complicated where privacy is concerned, because of the abundance of user information that can be found when correlated with information from users within the same social circle. For example, users can control which information they disseminate on their Facebook profile but they can’t control what information others will do so about them (Facebook Photo tagging is a good example of this).

As a result, novel privacy measure must be put into place. A worthwhile addition to SOSENS (and Social Sensing in general) would be exploring the application of well-known algorithms such as k-anonymity [138] in Social Sensing spaces. In the case of SOSENS, these algorithms could be added to the Data Parser ABB or to the Data Processing ABB (see Chapter 5), according to the level of privacy required.

## More applications consuming Semantic data

All this semantically rich data is really only useful as long as there are applications designed to take advantage of it. An interesting direction for future work is designing “*experiment-agnostic*” Social Sensing applications, where the data can be aggregated from a multitude of datasets. For example, a “Generic visualisation application for Social Sensing environments” could be built, which can produce screens such as **Figure 46**.



This application uses `foaf:knows` properties to calculate mutual friends between two `foaf:Person` and retrieves the most recent `sosens-t:Rating` for each one (for a measuring capability of the user's choice – the mBRS Rating for all Exercises calculated in the Pilot implementation is shown here for reference). It then visualizes them; each person is a node, the Rating value determines the node label and size and the mutual friends between the nodes determine the edge label and weight.

Such applications are very useful for drawing conclusions in Social Sensing spaces, especially for non-Computer-Science researchers. The data that powers such applications can be aggregated from a multitude of datasets and one SOSENS-based dataset can directly augment another. For example, the results of two Social Sensing experiments for exercise monitoring by different research teams on different Social Networking Sites (e.g. Facebook and Twitter) can be merged automatically, if all data is filtered through SOSENS.

## Promotion & Adoption

Finally, there are plans to promote the adoption of the SOSENS Framework by contacting relevant stakeholders (i.e. research groups that conduct Social Sensing

experiments) for testing in different use cases. An initial outreach in the local universities that participated in the NutriHeAl project has already been responded to with great interest, as has a potential collaboration with European researchers on a Weather Crowdsourcing & Social Sensing platform.

It is my firm belief that, as Semantic Web Technologies and frameworks like SOSENS mature, more applications will be built to take advantage of the rich semantic data they provide, which will be a benefit for Social Sensing in many scientific disciplines, especially in the fields of Web Science, Open Science & Linked Research.





# REFERENCES

- [1] M. F. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.
- [2] D. Ferrucci, "Build Watson: an overview of DeepQA for the Jeopardy! challenge," in *Proceedings of the 19th international conference on Parallel architectures and compilation techniques*, 2010, pp. 1–2.
- [3] M. Srivastava, T. Abdelzaher, B. Szymanski, M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric sensing," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 176–197, 2012.
- [4] S. Shiffman, A. A. Stone, and M. R. Hufford, "Ecological momentary assessment," *Annu Rev Clin Psychol*, vol. 4, pp. 1–32, 2008.
- [5] F. B. Hu *et al.*, "Reproducibility and validity of dietary patterns assessed with a food-frequency questionnaire," *The American Journal of Clinical Nutrition*, vol. 69, no. 2, pp. 243–249, 1999.
- [6] W. C. Willett *et al.*, "Reproducibility and validity of a semiquantitative food frequency questionnaire," *American journal of epidemiology*, vol. 122, no. 1, pp. 51–65, 1985.
- [7] A. G. Bonomi and K. R. Westerterp, "Advances in physical activity monitoring and lifestyle interventions in obesity: a review," *International Journal of Obesity*, vol. 36, no. 2, pp. 167–177, 2011.
- [8] M. Bishr and K. Janowicz, "Can we trust information?-the case of volunteered geographic information," in *Towards Digital Earth Search Discover and Share Geospatial Data Workshop at Future Internet Symposium, volume*, 2010, vol. 640.
- [9] A. Sheth, "Citizen sensing, social signals, and enriching human experience," *IEEE Internet Computing*, vol. 13, no. 4, p. 87, 2009.
- [10] "Users of the world, unite! The challenges and opportunities of Social Media." [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0007681309001232>. [Accessed: 31-Oct-2016].
- [11] D. M. Boyd and N. B. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, Oct. 2007.
- [12] "Alexa.com - Facebook.com Site Info." [Online]. Available: <http://www.alexa.com/siteinfo/facebook.com>. [Accessed: 01-Mar-2012].
- [13] "Facebook users worldwide 2016," *Statista*. [Online]. Available: <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>. [Accessed: 31-Oct-2016].
- [14] A. Karimi and C. Neustaedter, "My grandma uses facebook: Communication practices of older adults in an age of social media," Technical Report 2011–1116-04, 2011.
- [15] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: real-time event detection by social sensors," in *Proceedings of the 19th*

- international conference on World wide web*, New York, NY, USA, 2010, pp. 851–860.
- [16] A. Rosi, M. Mamei, F. Zambonelli, S. Dobson, G. Stevenson, and Juan Ye, “Social sensors and pervasive services: Approaches and perspectives,” in *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2011, pp. 525–530.
  - [17] N. A. Christakis and J. H. Fowler, “Social Network Sensors for Early Detection of Contagious Outbreaks,” *PLoS ONE*, vol. 5, no. 9, p. e12948, 2010.
  - [18] D. Wang, T. Abdelzaher, and L. Kaplan, “Chapter 2 - Social sensing trends and applications,” in *Social Sensing*, Boston: Morgan Kaufmann, 2015, pp. 13–20.
  - [19] B. Hull *et al.*, “CarTel: a distributed mobile sensor computing system,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*, 2006, pp. 125–138.
  - [20] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, “Cooperative transit tracking using smart-phones,” in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, 2010, pp. 85–98.
  - [21] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, “BikeNet: A mobile sensing system for cyclist experience mapping,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 1, p. 6, 2009.
  - [22] B. Resch, “People as Sensors and Collective Sensing-Contextual Observations Complementing Geo-Sensor Network Measurements,” in *Progress in Location-Based Services*, J. M. Krisp, Ed. Springer Berlin Heidelberg, 2013, pp. 391–406.
  - [23] H.-U. Prokosh, “Out with the old, in with the new: recycling data,” *HITE*, vol. 4, no. 1, 2011.
  - [24] P. S. Earle, D. C. Bowden, and M. Guy, “Twitter earthquake detection: Earthquake monitoring in a social world,” *Annals of Geophysics*, vol. 54, no. 6, p. 708715, 2011.
  - [25] Y. Qu, C. Huang, P. Zhang, and J. Zhang, “Microblogging After a Major Disaster in China: A Case Study of the 2010 Yushu Earthquake,” in *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, New York, NY, USA, 2011, pp. 25–34.
  - [26] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, “Microblogging during two natural hazards events: what twitter may contribute to situational awareness,” 2010, p. 1079.
  - [27] S. Zhao, L. Zhong, J. Wickramasuriya, and V. Vasudevan, “Human as Real-Time Sensors of Social and Physical Events: A Case Study of Twitter and Sports Games,” *arXiv:1106.4300 [physics]*, Jun. 2011.
  - [28] M. Motoyama, B. Meeder, K. Levchenko, G. M. Voelker, and S. Savage, “Measuring Online Service Availability Using Twitter,” in *Proceedings of the 3rd Wconference on Online Social Networks*, Berkeley, CA, USA, 2010, pp. 13–13.
  - [29] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.

- [30] “List of citizen science projects,” *Wikipedia*. 13-Nov-2016.
- [31] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu, “Predicting Flu Trends using Twitter data,” in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 702–707.
- [32] J. Li and C. Cardie, “Early stage influenza detection from twitter,” *arXiv preprint arXiv:1309.7340*, 2013.
- [33] V. Lampos and N. Cristianini, “Tracking the flu pandemic by monitoring the social web,” in *2010 2nd International Workshop on Cognitive Information Processing*, 2010, pp. 411–416.
- [34] R. McCreadie *et al.*, “SUPER: Towards the use of Social Sensors for Security Assessments and Proactive Management of Emergencies,” in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 1217–1220.
- [35] S. Diplaris *et al.*, “SocialSensor: sensing user generated input for improved media discovery and experience,” in *Proceedings of the 21st international conference companion on World Wide Web*, 2012, pp. 243–246.
- [36] J. Ye *et al.*, “Semantic web technologies in pervasive computing: A survey and research roadmap,” *Pervasive and Mobile Computing*, vol. 23, pp. 1–25, Oct. 2015.
- [37] “Facebook Graph API.” [Online]. Available: <https://developers.facebook.com/docs/graph-api>.
- [38] T. Berners-Lee, J. Hendler, O. Lassila, and others, “The semantic web,” *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.
- [39] “W3C Semantic Web Activity Homepage.” [Online]. Available: <https://www.w3.org/2001/sw/>. [Accessed: 06-Nov-2016].
- [40] C. A. Yeung, I. Liccardi, K. Lu, O. Seneviratne, and T. Berners-Lee, “Decentralization: The future of online social networking,” in *W3C Workshop on the Future of Social Networking Position Papers*, 2009, vol. 2, pp. 2–7.
- [41] “Everywhere and nowhere,” *The Economist*, 19-Mar-2008.
- [42] “W3C Data Activity - Building the Web of Data.” [Online]. Available: <https://www.w3.org/2013/data/>. [Accessed: 06-Nov-2016].
- [43] “Semantic Web - W3C.” [Online]. Available: <https://www.w3.org/standards/semanticweb/>. [Accessed: 23-Nov-2016].
- [44] L. Yu, *A Developer’s Guide to the Semantic Web*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [45] “Resource Description Framework (RDF) Model and Syntax Specification.” [Online]. Available: <https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>. [Accessed: 08-Nov-2016].
- [46] “Notation 3 Logic.” [Online]. Available: <https://www.w3.org/DesignIssues/Notation3>. [Accessed: 08-Nov-2016].
- [47] “Turtle - Terse RDF Triple Language.” [Online]. Available: <https://www.w3.org/TeamSubmission/turtle/>. [Accessed: 08-Nov-2016].
- [48] “RDF Schema 1.1.” [Online]. Available: <https://www.w3.org/TR/rdf-schema/>. [Accessed: 08-Nov-2016].
- [49] “OWL Web Ontology Language Guide.” [Online]. Available: <http://www.w3.org/TR/owl-guide/>. [Accessed: 30-Sep-2014].

- [50] “An introduction to the Semantic Web for health sciences librarians.” [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1435839/>. [Accessed: 08-Nov-2016].
- [51] M. Krötzsch, “OWL 2 Profiles: An introduction to lightweight ontology languages,” in *Reasoning Web International Summer School*, 2012, pp. 112–183.
- [52] “OWL 2 Web Ontology Language Document Overview (Second Edition).” [Online]. Available: <https://www.w3.org/TR/owl2-overview/>. [Accessed: 08-Nov-2016].
- [53] “OWL 2 Web Ontology Language Profiles (Second Edition).” [Online]. Available: <https://www.w3.org/TR/owl2-profiles/>. [Accessed: 09-Nov-2016].
- [54] D. Brickley and L. Miller, “FOAF vocabulary specification 0.98,” *Namespace document*, vol. 9, 2012.
- [55] G. Niezen, “Ontologies for interaction: Enabling serendipitous interoperability in smart environments,” *Journal of Ambient Intelligence and Smart Environments*, vol. 5, no. 1, pp. 135–137, Jan. 2013.
- [56] I. Oliver and J. Honkola, “Personal semantic web through a space based computing environment,” *arXiv preprint arXiv:0808.1455*, 2008.
- [57] “OWL/Implementations - Semantic Web Standards.” [Online]. Available: <https://www.w3.org/2001/sw/wiki/OWL/Implementations>. [Accessed: 09-Nov-2016].
- [58] “SPARQL Query Language for RDF.” [Online]. Available: <http://www.w3.org/TR/rdf-sparql-query/>.
- [59] “SPARQL Query Language for RDF - Solution Modifiers.” [Online]. Available: <https://www.w3.org/TR/rdf-sparql-query/#solutionModifiers>. [Accessed: 11-Nov-2016].
- [60] “SPARQL 1.1 Overview.” [Online]. Available: <https://www.w3.org/TR/sparql11-overview/>. [Accessed: 11-Nov-2016].
- [61] “Why SPARQL? - TechnicaLee Speaking.” [Online]. Available: [http://www.thefigtrees.net/lee/blog/2008/01/why\\_sparql.html](http://www.thefigtrees.net/lee/blog/2008/01/why_sparql.html). [Accessed: 11-Nov-2016].
- [62] LeeFeigenbaum, “SPARQL Cheat Sheet,” 01:12:58 UTC.
- [63] C. Bizer, T. Heath, and T. Berners-Lee, “Linked data-the story so far,” *International journal on semantic web and information systems*, vol. 5, no. 3, pp. 1–22, 2009.
- [64] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “DBpedia: A Nucleus for a Web of Open Data,” in *SpringerLink*, Springer Berlin Heidelberg, pp. 722–735.
- [65] “Semantic Web - Implications for Technologies and Business | Michael Workman | Springer.” [Online]. Available: <http://www.springer.com/gp/book/9783319166575>. [Accessed: 11-Nov-2016].
- [66] admin, “Introduction to the Semantic Web,” *Cambridge Semantics*, 17-Nov-2014. [Online]. Available: <http://www.cambridgesemantics.com/semantic-university/introduction-semantic-web>. [Accessed: 11-Nov-2016].

- [67] “A Direct Mapping of Relational Data to RDF.” [Online]. Available: <https://www.w3.org/TR/2012/REC-rdb-direct-mapping-20120927/>. [Accessed: 06-Nov-2016].
- [68] P. Mika, *Social Networks and the Semantic Web*, vol. 5. 2007.
- [69] “W3C Semantic Web Interest Group (SWIG).” [Online]. Available: <https://www.w3.org/2001/sw/interest/>. [Accessed: 06-Nov-2016].
- [70] M. Compton *et al.*, “The SSN Ontology of the Semantic Sensor Networks Incubator Group,” *Journal of Web Semantics*, 2011.
- [71] K. M. Polzien, J. M. Jakicic, D. F. Tate, and A. D. Otto, “The Efficacy of a Technology-based System in a Short-term Behavioral Weight Loss Intervention,” *Obesity*, vol. 15, no. 4, pp. 825–830, 2012.
- [72] Oscar Corcho, “Semantic (Social) Sensor Networks,” 9th Summer School on Ontological Engineering and the Semantic Web, 14-Jul-2012.
- [73] “Ontology:DOLCE+DnS Ultralite - Odp.” [Online]. Available: [http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS\\_Ultralite](http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite). [Accessed: 05-Oct-2016].
- [74] A. Sheth, C. Henson, and S. S. Sahoo, “Semantic sensor web,” *Internet Computing, IEEE*, vol. 12, no. 4, pp. 78–83, 2008.
- [75] “SSN Applications - Semantic Sensor Networks Community Group.” [Online]. Available: [https://www.w3.org/community/ssn-cg/wiki/SSN\\_Applications](https://www.w3.org/community/ssn-cg/wiki/SSN_Applications). [Accessed: 05-Oct-2016].
- [76] “Spatial Data on the Web Working Group.” [Online]. Available: [https://www.w3.org/2015/spatial/wiki/Main\\_Page](https://www.w3.org/2015/spatial/wiki/Main_Page). [Accessed: 05-Oct-2016].
- [77] “Semantic Sensor Network XG Final Report.” [Online]. Available: <http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/>. [Accessed: 15-Jan-2013].
- [78] “European Commission: CORDIS: Proactive Knowledge for Smart Products.” [Online]. Available: [http://cordis.europa.eu/project/rcn/89484\\_en.html](http://cordis.europa.eu/project/rcn/89484_en.html). [Accessed: 05-Oct-2016].
- [79] “MUO Ontology.” [Online]. Available: <http://idi.fundacionctic.org/muo/>. [Accessed: 05-Oct-2016].
- [80] G. Erétéo *et al.*, “Semantic Social Network Analysis: A Concrete Case,” *Handbook of Research on Methods and Techniques for Studying Virtual Communities: Paradigms and Phenomena*, pp. 122–156, 2011.
- [81] J. Golbeck and M. Rothstein, “Linking social networks on the web with FOAF: a semantic web case study,” in *Proceedings of the 23rd national conference on Artificial intelligence*, 2008, vol. 2, pp. 1138–1143.
- [82] “RELATIONSHIP: A vocabulary for describing relationships between people.” [Online]. Available: <http://vocab.org/relationship/.html>. [Accessed: 15-Jan-2013].
- [83] “Descriptions of Social Relations.” [Online]. Available: <http://www.cs.vu.nl/~pmika/research/foaf-ws/foaf-x.html>. [Accessed: 16-Apr-2013].
- [84] “FoafExtensions - FOAF Wiki.” [Online]. Available: <http://wiki.foaf-project.org/w/FoafExtensions>. [Accessed: 15-Jan-2013].

- [85] “SIOC Core Ontology Specification.” [Online]. Available: <http://rdfs.org/sioc/spec/>. [Accessed: 09-Oct-2016].
- [86] T. Lebo *et al.*, “Prov-o: The prov ontology,” *W3C Recommendation*, 30th April, 2013.
- [87] K. N. Vavliakis, A. L. Symeonidis, and P. A. Mitkas, “Event identification in web social media through named entity recognition and topic modeling,” *Data & Knowledge Engineering*, vol. 88, pp. 1–24, 2013.
- [88] A. Vakali and K. Kafetsios, “Emotion aware clustering analysis as a tool for Web 2.0 communities detection: Implications for curriculum development,” in *World Wide Web Conference, WWW*, 2012.
- [89] M. Sabou, J. Kantorovitch, A. Nikolov, A. Tokmakoff, X. Zhou, and E. Motta, “Position paper on realizing smart products: Challenges for semantic web technologies,” in *Proceedings of the 2nd International Conference on Semantic Sensor Networks-Volume 522*, 2009, pp. 135–147.
- [90] D. Artz and Y. Gil, “A survey of trust in computer science and the Semantic Web,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 58–71, Jun. 2007.
- [91] D. Wang *et al.*, “Using humans as sensors: An estimation-theoretic perspective,” in *Proceedings of the 13th international symposium on Information processing in sensor networks*, 2014, pp. 35–46.
- [92] C. Fürber and M. Hepp, “Using Semantic Web Technologies for Data Quality Management,” in *Handbook of Data Quality*, S. Sadiq, Ed. Springer Berlin Heidelberg, 2013, pp. 141–161.
- [93] J. Golbeck, “Weaving a web of trust,” *Science*, vol. 321, no. 5896, pp. 1640–1641, 2008.
- [94] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques*. Springer Science & Business Media, 2006.
- [95] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, Mar. 2007.
- [96] “Integrity Constraints in OWL,” *Tetherless World Constellation*. [Online]. Available: [https://tw.rpi.edu/web/doc/integrity\\_constraints\\_in\\_owl](https://tw.rpi.edu/web/doc/integrity_constraints_in_owl). [Accessed: 28-Sep-2016].
- [97] B. K. Kahn, D. M. Strong, and R. Y. Wang, “Information quality benchmarks: product and service performance,” *Communications of the ACM*, vol. 45, no. 4, pp. 184–192, 2002.
- [98] A. Russell and P. Smart, “Nitelight: A graphical editor for sparql queries,” 2008.
- [99] O. Ambrus, K. Möller, and S. Handschuh, “Konduit VQB: a Visual Query Builder for SPARQL on the Social Semantic Desktop,” in *Workshop on Visual Interfaces to the Social and Semantic Web*, 2010.
- [100] “Ontotext GraphDB™.” [Online]. Available: <http://ontotext.com/products/ontotext-graphdb/>.
- [101] “Facebook: monthly active users 2015 | Statista,” *Statista*. [Online]. Available: <http://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>. [Accessed: 23-Apr-2015].

- [102] M. Duggan, N. B. Ellison, C. Lampe, Am, a Lenhart, and M. Madden, "Demographics of Key Social Networking Platforms," *Pew Research Center: Internet, Science & Tech.* .
- [103] L. R. Buis, "The potential for web-based social network sites and self-regulation for health promotion," *Am J Health Promot*, vol. 26, no. 2, pp. 73–76, Dec. 2011.
- [104] I. Pagkalos and L. Petrou, "Using Social Network Apps as Social Sensors for Health Monitoring," presented at the MEDICON 2013, Seville, Spain, 2013.
- [105] I. Pagkalos and M. Hassapidou, "Using social applications to complement physical activity monitoring," *Obes Facts*, vol. 6, no. s1, p. 31, 2013.
- [106] I. Pagkalos, D. Rossiou, and S. Papadopoulou, "Monitoring physical activity through social networks: a Facebook case study," presented at the 6th DIETS/EFAD Conference, Portoroz, Slovenia, 2012.
- [107] W. N. Schofield, "Predicting basal metabolic rate, new standards and review of previous work," *Hum Nutr Clin Nutr*, vol. 39 Suppl 1, pp. 5–41, 1985.
- [108] "WHO | The world health report 2000 - Health systems: improving performance," *WHO*. [Online]. Available: <http://www.who.int/whr/2000/en/>. [Accessed: 28-Sep-2015].
- [109] B. E. Ainsworth *et al.*, "2011 compendium of physical activities: a second update of codes and MET values," *Medicine and Science in Sports and Exercise*, vol. 43, no. 8, pp. 1575–1581, 2011.
- [110] J. A. Noah, D. K. Spierer, J. Gu, and S. Bronner, "Comparison of steps and energy expenditure assessment in adults of Fitbit Tracker and Ultra to the Actical and indirect calorimetry," *Journal of Medical Engineering & Technology*, vol. 37, no. 7, pp. 456–462, Oct. 2013.
- [111] K. M. Diaz *et al.*, "Abstract MP11: Fitbit: An Accurate and Reliable Device for Wireless Physical Activity Tracking," *Circulation*, vol. 131, no. Suppl 1, pp. AMP11-AMP11, Mar. 2015.
- [112] M. A. Tully, C. McBride, L. Heron, and R. F. Hunter, "The validation of Fitbit Zip™ physical activity monitor as a measure of free-living physical activity," *BMC Research Notes*, vol. 7, no. 1, p. 952, Dec. 2014.
- [113] S. J. Marshall *et al.*, "Translating physical activity recommendations into a pedometer-based step goal: 3000 steps in 30 minutes," *Am J Prev Med*, vol. 36, no. 5, pp. 410–415, May 2009.
- [114] D. M. Harrington, K. P. Dowd, C. Tudor-Locke, and A. E. Donnelly, "A steps/minute value for moderate intensity physical activity in adolescent females," *Pediatr Exerc Sci*, vol. 24, no. 3, pp. 399–408, Aug. 2012.
- [115] J. D. Pillay, T. L. Kolbe-Alexander, K. I. Proper, W. van Mechelen, and E. V. Lambert, "Steps that count: physical activity recommendations, brisk walking, and steps per minute--how do they relate?," *J Phys Act Health*, vol. 11, no. 3, pp. 502–508, Mar. 2014.
- [116] "17 - Walking - Compendium of Physical Activities." [Online]. Available: <https://sites.google.com/site/compendiumofphysicalactivities/Activity-Categories/walking>. [Accessed: 29-Sep-2015].

- [117] S. W. Lichtman *et al.*, “Discrepancy between self-reported and actual caloric intake and exercise in obese subjects,” *New England Journal of Medicine*, vol. 327, no. 27, pp. 1893–1898, 1992.
- [118] L. A. Zadeh, “Fuzzy sets,” *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.
- [119] B. Bouchon-Meunier, M. Dotoli, and B. Maione, “On the choice of membership functions in a mamdani-type fuzzy controller,” 1996.
- [120] B. E. Commerce, A. Jøsang, and R. Ismail, “The beta reputation system,” in *In Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.
- [121] A. Jøsang and J. Haller, “Dirichlet reputation systems,” in *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, 2007, pp. 112–119.
- [122] L. R. Smeltzer, “The meaning and origin of trust in buyer-supplier relationships,” *International Journal of Purchasing and Materials Management*, vol. 33, no. 1, pp. 40–48, 1997.
- [123] S. Reddy, D. Estrin, and M. Srivastava, “Recruitment framework for participatory sensing data collections,” *Pervasive Computing*, pp. 138–155, 2010.
- [124] A. Jøsang and W. Quattrociocchi, *Advanced features in bayesian reputation systems*. Springer, 2009.
- [125] A. Jøsang, X. Luo, and X. Chen, “Continuous ratings in discrete bayesian reputation systems,” in *Trust Management II*, Springer, 2008, pp. 151–166.
- [126] T. van Deursen, P. Koster, and M. Petkovic, “Hedaquin: A Reputation-based Health Data Quality Indicator,” *Electronic Notes in Theoretical Computer Science*, vol. 197, no. 2, pp. 159–167, Feb. 2008.
- [127] B. Alhaqbani, A. Jøsang, and C. Fidge, “A medical data reliability assessment model,” *Journal of theoretical and applied electronic commerce research*, vol. 4, no. 2, pp. 64–78, 2009.
- [128] E. Kapantais *et al.*, “First National Epidemiological Survey on the Prevalence of Obesity and Abdominal Fat Distribution in Greek Adults,” *Annals of Nutrition and Metabolism*, vol. 50, no. 4, pp. 330–338, 2006.
- [129] L. W. Lacy, *OWL: Representing information using the web ontology language*. Trafford Publishing, 2005.
- [130] J. Scott, “Social network analysis,” *Sociology*, vol. 22, no. 1, pp. 109–127, 1988.
- [131] K. Blanchet and P. James, “How to do (or not to do)... a social network analysis in health systems research,” *Health policy and planning*, vol. 27, no. 5, pp. 438–446, 2012.
- [132] F. Griffiths *et al.*, “Social networks—The future for health care delivery,” *Social science & medicine*, vol. 75, no. 12, pp. 2233–2241, 2012.
- [133] “RIF - Semantic Web Standards.” [Online]. Available: <https://www.w3.org/2001/sw/wiki/RIF>. [Accessed: 16-Nov-2016].
- [134] “SWRL: A Semantic Web Rule Language Combining OWL and RuleML.” [Online]. Available: <https://www.w3.org/Submission/SWRL/>. [Accessed: 16-Nov-2016].



- [135] “SPIN - Overview and Motivation.” [Online]. Available: <https://www.w3.org/Submission/spin-overview/>. [Accessed: 16-Nov-2016].
- [136] C. Baillie, P. Edwards, and E. Pignotti, “QUAL: A Provenance-Aware Quality Model,” *Journal of Data and Information Quality (JDIQ)*, vol. 5, no. 3, p. 12, 2015.
- [137] C. Fürber and M. Hepp, “Swiqa-a semantic web information quality assessment framework.,” in *ECIS*, 2011, vol. 15, p. 19.
- [138] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.
- [139] T. R. Wójcicki, D. Grigsby-Toussaint, C. H. Hillman, M. Huhman, and E. McAuley, “Promoting Physical Activity in Low-Active Adolescents via Facebook: A Pilot Randomized Controlled Trial to Test Feasibility,” *JMIR Res Protoc*, vol. 3, no. 4, p. e56, 2014.
- [140] L. S. Lao, “Evaluation of a social networking based SNAP-Ed nutrition curriculum on behavior change,” *URI Dissertations and Master’s Theses (Campus Access)*, pp. 1–135, Jan. 2011.
- [141] C. G. Valle, D. F. Tate, D. K. Mayer, M. Allicock, and J. Cai, “A randomized trial of a Facebook-based physical activity intervention for young adult cancer survivors,” *Journal of Cancer Survivorship*, vol. 7, no. 3, pp. 355–368, 2013.
- [142] D. N. Cavallo, D. F. Tate, A. V. Ries, J. D. Brown, R. F. DeVellis, and A. S. Ammerman, “A social media-based physical activity intervention: a randomized controlled trial,” *Am J Prev Med*, vol. 43, no. 5, pp. 527–532, Nov. 2012.
- [143] “MyFitnessPal.” [Online]. Available: [https://www.facebook.com/games/myfitnesspal\\_fb/](https://www.facebook.com/games/myfitnesspal_fb/). [Accessed: 03-Nov-2015].
- [144] R. K. Dishman and J. Buckworth, “Increasing physical activity: a quantitative synthesis,” *Med Sci Sports Exerc*, vol. 28, no. 6, pp. 706–719, Jun. 1996.
- [145] L. Litman, Z. Rosen, D. Spierer, S. Weinberger-Litman, A. Goldschein, and J. Robinson, “Mobile Exercise Apps and Increased Leisure Time Exercise Activity: A Moderated Mediation Analysis of the Role of Self-Efficacy and Barriers,” *Journal of Medical Internet Research*, vol. 17, no. 8, p. e195, Aug. 2015.
- [146] D. Foster, C. Linehan, B. Kirman, S. Lawson, and G. James, “Motivating physical activity at work: using persuasive social media for competitive step counting,” in *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments*, 2010, pp. 111–116.
- [147] C. Maher *et al.*, “A Web-Based, Social Networking Physical Activity Intervention for Insufficiently Active Adults Delivered via Facebook App: Randomized Controlled Trial,” *J Med Internet Res*, vol. 17, no. 7, Jul. 2015.
- [148] J. Kernot, T. Olds, L. K. Lewis, and C. Maher, “Usability Testing and Piloting of the Mums Step It Up Program-A Team-Based Social Networking Physical Activity Intervention for Women with Young Children,” 2014.

- 
- [149] A. I. of Health and Welfare, *The Active Australia Survey: A guide and manual for implementation, analysis and reporting*. Australian Institute of Health and Welfare, 2003.
  - [150] D. Ding, S. Ayubi, S. Hiremath, and B. Parmanto, “Physical activity monitoring and sharing platform for manual wheelchair users,” in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, 2012, pp. 5833–5836.
  - [151] S. G. Trost, D. S. Ward, B. McGraw, and R. R. Pate, “Validity of the Previous Day Physical Activity Recall (PDPAR) in fifth-grade children,” *Pediatric Exercise Science*, vol. 11, no. 4, p. 341, 1999.
  - [152] R. G. McMurray *et al.*, “Comparison of two approaches to structured physical activity surveys for adolescents,” *Medicine and science in sports and exercise*, vol. 36, no. 12, p. 2135, 2004.
  - [153] P. Z. Siegel, R. M. Brackbill, and G. W. Heath, “The epidemiology of walking for exercise: implications for promoting activity among sedentary groups,” *American journal of public health*, vol. 85, no. 5, pp. 706–710, 1995.

# APPENDIX A

## The NutriHeAI Pilot

### Background

Facebook *Groups* (dedicated, potentially closed spaces that facilitate content sharing between group members<sup>48</sup>) have been used in the past for intervention programs that promoted & monitored physical activity, by uploading relevant information and resources as group “wall posts” and/or collecting data by encouraging users to answer self-reported questionnaires (e.g. [139]–[142]).

While Facebook groups are an easy, accessible choice for SNS-based e-Health research, Facebook also offers a very well documented, free and versatile platform for *application development*<sup>49</sup> which allows researchers to provide custom content and easily benefit from the existence of both social data and user-generated content in the same platform. A Web developer can build an application that could be as simple as an HTML Web Form accessed from Facebook and offer it to the public or a selected audience. In addition, the developer can specify whether to request parts of the users’ data that exists on the platform (social data, likes, interests etc.). Over 1 million users use health & fitness SNApps such as MyFitnessPal [143], which aids users in keeping a food and exercise journal, among other features. A recent study showed that exercise (mobile) app users are more likely to exercise during their leisure time (one of the most important times for exercise [144]), compared to those who do not use exercise apps [145].

Even though Facebook Apps are an established staple of the popular SNS and the amount of potential users is extremely large, to date only a handful of peer-reviewed studies have explicitly used one for exercise monitoring. Foster et al [146] use a Facebook app where participants (10 co-workers in a UK hospital) self-report their daily step count. The same concept of daily steps self-reporting is also used in Maher et al. [147], where 110 adults (mean age 35.6 years) participated in an intervention for insufficiently active adults via Facebook. The “Mums Step It Up” [148] program in Australia, aimed at mothers with young children, also tracks daily steps via a Facebook app throughout a 28 days period and assesses physical activity by distributing the Active Australia physical activity questionnaire [149]. Ding et al [150] developed a physical activity monitoring and sharing platform

---

<sup>48</sup> Facebook groups: <https://www.facebook.com/help/284236078342160>

<sup>49</sup> Facebook for Developers: <https://developers.facebook.com/>

(PAMS) for manual wheelchair users where a Facebook app was used for monitoring and sharing users' progress, as reported by a monitoring unit installed on the wheelchair. It is worthwhile to note that in all the above studies, daily physical activity levels were increased.

## Pilot Results

The main body of the thesis described the methodology, design and implementation of the NutriHeAl ESS Pilot in regards to the SOSENS framework. Designed to run for a period of 5 weeks, a large percentage of users successfully participated, creating a variety of data that, through SOSENS, is machine-understandable and easily re-usable. The section that follows presents & discusses the results of the pilot in more detail as well as from an e-Health viewpoint.

## Sensory Data Collected

A total of 44 individuals (dropout of 5, ~10%) completed the pilot. Out of the 35 days (5 weeks) of the project's duration, activities were reported, on average, for  $33 \pm 5$  days (where  $\pm$  the Standard Deviation).

Activity Category	Total	% of total	AVG / User
Moderate+	586	10%	$13 \pm 16$
Low+	1024	17%	$23 \pm 23$
Work	502	8%	$11 \pm 11$
Sleep	1355	23%	$31 \pm 10$
Other	2461	42%	$56 \pm 60$
<b>TOTAL</b>	<b>5928</b>		<b><math>135 \pm 85</math></b>
<b>Table 6:</b> Activities recorded by category, and relevant statistics			

The NutriHeAl app collected data for, approximately, 6000 activities. **Table 6** shows an overall view of the data (all activities) while **Table 7** focuses on exercise activities only, which was the intended focus of the pilot.

Users reported a total of 1610 exercises (along with a large number of other activities which were recorded but not processed during this pilot study) of which 1024 (64%) belonged to the Low+ category and 586 (36%) to the Moderate+ category. On average, each user submitted  $37 \pm 29$  activities of which  $23 \pm 23$  were in the Low+ category and  $13 \pm 16$  on the Moderate+ category.

Exercise Category	Total	%of total	AVG / user
<b>Low+</b>	1024	64%	23±23
<b>Moderate+</b>	586	36%	13±16
<b>TOTAL</b>	<b>1610</b>		37±29
<i>Popular Low+ Activities:</i> Walking (95%) <i>Popular Moderate+ Activities:</i> Gym (18%), Bicycle (16%), Dancing (13%), Running (12%)			
<b>Table 7:</b> Exercises recorded by MET category and popular activities			

The group's mean time of submission was  $40 \pm 43$  hours after each activity. More specifically, 20% of the users reported the activity within 12 hours of its reported end, 27% within 24 hours, 25% within 48 hours and 27% after 48 hours had passed.

**Figure 47** shows a detailed view of the volume of data collected per user

The *public profiles & social circles* (friends) of the users were also downloaded, with users having an average of  $447 \pm 361$  social connections. No users rejected the requested "friends" permission.

*Fitbit data* was provided by the users for the entirety of the pilot's duration.

## Evaluation Results

As discussed in section 5.3, all user-submitted exercise activities were evaluated against the membership functions for each exercise category (Low+ or Medium+). Table 2 shows three metrics computed from these evaluations:

- EV1: Exercise Reporting Accuracy: Each user's exercise activities (independent of category) were evaluated and averaged to compute a user's Exercise Reporting Accuracy (ERA) score. Afterwards, users' ERAs were averaged to compute the group's ERA.
- EV2: Low+ Exercise Reporting Accuracy: As above, but for Low+ exercise activities only
- EV3: Moderate+ Exercise Reporting Accuracy: As above, but for Moderate+ exercise activities only.

Variable	Group Total
EV1: Exercise Reporting Accuracy (All Exercise)	71 $\pm$ 21%
EV2: Low+ Exercise Reporting Accuracy	82 $\pm$ 18%
EV3: Moderate+ Exercise Reporting Accuracy	51 $\pm$ 31%

**Table 8:** Reporting accuracy by evaluation metric

## Discussing the results of the pilot

As can be seen from the low dropout (10%) and the high number of days with recorded activities (33 $\pm$ 5), reporting activities via the Facebook app was an effective way of data collection. A possible contributing factor to this is the free-form text entry in combination with listing the “recent” and “previously-submitted” activities which has aided users in consistently providing their reports over the pilot’s duration. Another important contributing factor, and the reason why SNApPs are a promising research tool, is the fact that users spend a lot of time on Facebook for their own reasons, and interacting with an app within the same environment is probably not considered a distraction. While using the app, the user has access to chat, notifications and other Facebook aspects, which helps to create the idea that he/she does not exit the platform to use the app.

Users had varying reporting habits, but most reported their activity within 1 (47% of user base) or 2 (72% of user base) days of its completion. Only a handful of users (10%) reported their activities within a few hours, which is to be expected, as the motivation for each user was seeing the graphs at the end of each week. Still, their reporting frequency mimics 1-day and 3-day physical activity recall questionnaires (such as PDPAR [151] and 3DPAR [152]) which have been shown to be a valid method for physical activity recall. Out of the reported activities, two thirds were categorised as Low+ exercises, of which the vast majority (~95%) were walking activities. Seeing that walking is widely reported as the most common form of physical activity [153], this is an expected conclusion that reflects a healthy sample.

Low+ activities were also the activity group with the highest ERA (82 $\pm$ 18%) which shows that users could, in large, accurately assess activities such as walking. The large variance in Moderate+ ERA scores (51  $\pm$  31%) can possibly be attributed to the lower number of such activities present in the sample, in comparison to the Low+ group. Some users reported only Low+ activities while others reported both. In addition, it should be noted that using 1 activity *mf* for each activity category is not optimal and, ideally, each different activity should have a unique *mf*. Given the fact that research that correlates step counts to individual activities is limited, this was a best-effort approach.

In regards to ERAs over time, it is possible that viewing the results each week could influence the users and improve their accuracy over time, but after calculating each user's mean accuracy score per cycle (7 days) there was no conclusive evidence that pointed towards a statistical correlation between the weeks in the study and the accuracy score (not significant at  $p < 0.05$ ). Seeing that the sample size (an average of 5 7-day cycles per user) is small, such an improvement may be apparent over a larger time span. The same notion was also explored for 3-day cycles but no definitive conclusion could be reached for that time span either.

#	H	W	S	Age	BMI	Steps /day	Steps /day ±	#reports	rep/day	AVG_TO S_Diff	# Mod+	% Mod+	Mod+ / day	# Low+	% Low+	Low+ / day	# Work	% Work	Work/ day	# Sleep	% Sleep	Sleep/ day	# Oth	% Oth	Oth / day	# Exer (L+M)	% Exer (L+M)	Exer /day
1	182	79	M	30	23.8	8599	4737	192	4	1680	41	21%	0.9	68	35%	1.5	21	11%	0.5	40	21%	0.9	22	11%	0.5	109	57%	2.4
2	163	65	F	22	24.5	5493	3325	162	5	5144	0	0%	0.0	4	2%	0.1	15	9%	0.5	33	20%	1.1	110	68%	3.5	4	2%	0.1
3	169	69	M	21	24.2	16407	7298	81	2	5436	5	6%	0.2	32	40%	1.0	2	2%	0.1	33	41%	1.0	9	11%	0.3	37	46%	1.1
4	175	80	F	24	26.1	2327	1867	45	2	9827	0	0%	0.0	29	64%	1.1	5	11%	0.2	7	16%	0.3	4	9%	0.1	29	64%	1.1
5	158	64	F	34	25.6	11083	4973	218	8	8901	32	15%	1.1	5	2%	0.2	12	6%	0.4	36	17%	1.3	133	61%	4.8	37	17%	1.3
6	164	62	F	20	23.1	3172	2785	58	2	2334	3	5%	0.1	0	0%	0.0	0	0%	0.0	30	52%	0.9	25	43%	0.7	3	5%	0.1
7	174	84	M	49	27.7	7065	3529	33	2	9799	0	0%	0.0	0	0%	0.0	26	79%	1.2	0	0%	0.0	7	21%	0.3	0	0%	0.0
8	171	74	M	24	25.3	7830	3613	95	3	1257	12	13%	0.3	6	6%	0.2	27	28%	0.8	34	36%	1.0	16	17%	0.5	18	19%	0.5
9	166	74	F	21	26.9	10880	5986	84	2	1187	14	17%	0.4	13	15%	0.3	0	0%	0.0	33	39%	0.8	24	29%	0.6	27	32%	0.7
10	179	80	M	21	25.0	13191	5331	67	3	2969	3	4%	0.1	12	18%	0.5	9	13%	0.4	23	34%	0.9	20	30%	0.8	15	22%	0.6
11	164	67	F	23	24.9	14417	8409	77	2	1172	28	36%	0.8	5	6%	0.1	0	0%	0.0	39	51%	1.1	5	6%	0.1	33	43%	0.9
12	170	53	F	21	18.3	11036	4220	75	2	2227	23	31%	0.6	8	11%	0.2	0	0%	0.0	41	55%	1.0	3	4%	0.1	31	41%	0.8
13	167	62	F	21	22.2	10678	4875	138	5	4557	16	12%	0.5	20	14%	0.7	15	11%	0.5	26	19%	0.9	61	44%	2.0	36	26%	1.2
14	160	55	F	20	21.5	16313	5029	79	2	1830	6	8%	0.2	39	49%	1.1	0	0%	0.0	34	43%	1.0	0	0%	0.0	45	57%	1.3
15	165	69	F	22	25.3	8262	4866	386	13	14484	15	4%	0.5	87	23%	2.9	2	1%	0.1	29	8%	1.0	253	66%	8.4	102	26%	3.4
16	160	56	F	22	21.9	11996	5426	81	2	3619	20	25%	0.6	38	47%	1.1	0	0%	0.0	20	25%	0.6	3	4%	0.1	58	72%	1.7
17	157	60	F	20	24.3	8132	4120	344	11	6591	91	26%	2.9	21	6%	0.7	47	14%	1.5	38	11%	1.2	147	43%	4.7	112	33%	3.6
18	173	65	F	20	21.7	9865	4728	78	2	1520	24	31%	0.7	4	5%	0.1	9	12%	0.3	32	41%	1.0	9	12%	0.3	28	36%	0.8
19	169	52	F	21	18.2	8255	5059	109	4	2461	14	13%	0.5	43	39%	1.5	10	9%	0.3	26	24%	0.9	16	15%	0.6	57	52%	2.0
20	160	50	F	21	19.5	8514	4390	133	4	1999	5	4%	0.2	14	11%	0.4	13	10%	0.4	33	25%	1.0	68	51%	2.1	19	14%	0.6
21	166	60	F	25	21.8	7380	4423	66	2	5791	13	20%	0.4	13	20%	0.4	14	21%	0.5	26	39%	0.8	0	0%	0.0	26	39%	0.8
22	169	60	F	22	21.0	8475	3504	88	3	4214	0	0%	0.0	4	5%	0.1	15	17%	0.5	34	39%	1.0	35	40%	1.1	4	5%	0.1
23	168	81	F	23	28.7	3193	1875	146	5	2944	1	1%	0.0	31	21%	1.0	11	8%	0.4	35	24%	1.1	68	47%	2.2	32	22%	1.0
24	163	62	F	23	23.3	10123	3679	51	1	2680	8	16%	0.2	11	22%	0.3	0	0%	0.0	32	63%	0.9	0	0%	0.0	19	37%	0.5
25	153	59	F	21	25.2	8785	4406	141	5	2339	6	4%	0.2	14	10%	0.5	21	15%	0.8	27	19%	1.0	73	52%	2.6	20	14%	0.7
26	164	53	F	21	19.7	7587	4162	151	5	3419	0	0%	0.0	24	16%	0.8	16	11%	0.6	30	20%	1.0	81	54%	2.8	24	16%	0.8
27	178	63	F	21	19.9	10160	4464	107	3	1351	21	20%	0.6	24	22%	0.7	21	20%	0.6	29	27%	0.8	12	11%	0.3	45	42%	1.3
28	186	73	M	41	21.1	7610	3274	291	8	1366	12	4%	0.3	92	32%	2.6	33	11%	0.9	37	13%	1.0	117	40%	3.3	104	36%	2.9
29	170	64	F	20	22.1	6727	3351	98	3	2106	17	17%	0.5	9	9%	0.3	24	24%	0.7	32	33%	0.9	16	16%	0.5	26	27%	0.8
30	179	80	M	40	25.0	7935	1868	204	6	2349	2	1%	0.1	5	2%	0.1	26	13%	0.7	37	16%	1.0	134	66%	3.7	7	3%	0.2
31	158	53	F	21	21.2	9526	4521	100	3	828	14	14%	0.4	24	24%	0.7	0	0%	0.0	33	33%	1.0	29	29%	0.9	38	38%	1.1
32	171	83	F	21	28.4	4860	2702	223	6	2842	0	0%	0.0	3	1%	0.1	21	9%	0.6	36	16%	1.0	163	73%	4.5	3	1%	0.1
33	158	50	F	22	20.0	8592	4424	147	4	2408	5	3%	0.1	14	10%	0.4	3	2%	0.1	38	26%	1.1	87	59%	2.6	19	13%	0.6
34	164	54	F	22	20.1	11938	6584	292	10	1771	6	2%	0.2	60	21%	2.0	16	5%	0.5	35	12%	1.2	175	60%	5.8	66	23%	2.2
35	155	50	F	22	20.8	9892	3503	152	4	1948	0	0%	0.0	33	22%	0.9	0	0%	0.0	45	30%	1.3	74	49%	2.1	33	22%	0.9
36	169	66	F	43	23.1	9862	4776	40	2	1533	6	15%	0.3	8	20%	0.4	13	33%	0.6	0	0%	0.0	13	33%	0.6	14	35%	0.6
37	170	55	F	30	19.0	7870	3256	76	2	1881	14	18%	0.4	11	14%	0.3	0	0%	0.0	33	43%	0.9	18	24%	0.5	25	33%	0.7
38	157	52	F	24	21.1	7377	3535	55	2	3844	0	0%	0.0	12	22%	0.4	7	13%	0.2	27	49%	0.9	9	16%	0.3	12	22%	0.4
39	180	76	M	23	23.5	12196	8636	112	3	1517	31	28%	0.9	12	11%	0.3	9	8%	0.3	36	32%	1.0	24	21%	0.7	43	38%	1.2
40	160	45	F	20	17.6	8681	3963	178	5	1345	23	13%	0.7	24	13%	0.7	4	2%	0.1	44	25%	1.3	83	47%	2.4	47	26%	1.4
41	158	51	F	23	20.4	7993	2715	323	8	2600	11	3%	0.3	86	27%	2.0	32	10%	0.7	28	9%	0.7	166	51%	3.9	97	30%	2.3
42	173	53	F	20	17.7	10098	4265	97	3	1716	15	15%	0.4	22	23%	0.6	3	3%	0.1	37	38%	1.1	20	21%	0.6	37	38%	1.1
43	158	50	F	20	20.0	7535	4023	107	3	759	27	25%	0.8	12	11%	0.3	0	0%	0.0	38	36%	1.1	30	28%	0.9	39	36%	1.1
44	162	46	F	21	17.5	8016	4127	148	6	3936	2	1%	0.1	28	19%	1.1	0	0%	0.0	19	13%	0.8	99	67%	4.0	30	20%	1.2
AVG	167	63		24	22.5	8971		135	4	3329	13			23			11			31			56			37		
±	8	11		7	2.96	2900		85	3	2809	16			23			11			10			60			29		
TOT								5928			586	10%		1024	17%		502	8%		1355	23%		2461	42%		1610	27%	

**Figure 47:**  
Data collected  
from users of  
the NutriHeAl  
App



# APPENDIX B

## The SOSENS API

This appendix provides a sample from the SOSENS API documentation for the `api/v1/observation` endpoint (i.e. converting new sensor observations to the SOSENS format):

### DEPENDENCIES

If you'd like to serve the SOSENS API, contact the author (ipagkalo at auth.gr) for the source code and make sure you have the dependencies installed by issuing the following command (here: using the `pip` Python Package installer – you can use whatever you like):

```
pip install python-dateutil rdflib flask-cors
```

### JSON API FORMAT

You should post to the API using the following generalisation (data & global field values explained within this document)

```
{
  "data": [
    { ... },
    { ... },
    { ... }
  ],
  "global": { ... }
}
```

## USE CASE #1: SIMPLE Mode

- If all you want to do is to simply create a few observations in SOSENS using the SOSENS defaults, just fill in the “required” type variables.
- The SOSENS API will create a URI for the observation in the form of {observedBy}\_obs\_{property}\_{time} and create the necessary RDF
- It will also handle the required property values according to valueType, using the SOSENS Generic SSN Value and the observation time using the TIME ontology.

JSON: data			
type	variable	example	comment
required	observedBy	Sensor_1	The sensor that created the observation (an OWL Instance)
required	feature	Area_1	The feature of interest that the sensor observes (an OWL Instance)
required	property	temperature	The observed property of the sensor (an OWL Instance)
required	time	20/05/2014 14:30	The time of the observation, in many acceptable formats (see <a href="http://bit.ly/2g34UfL">http://bit.ly/2g34UfL</a> )
required	valueType	quantity	Choose “quantity” or “quality”. This will use the SOSENS Generic SSN Value ontology to describe the observationValue.
required	value	50	The observed value of the observation (value only)
required	valueDataType	integer	Corresponds to XMLSchema values (double, integer, etc.)
optional	unitOfMeasurement	Celsius	The unit in which the observation value is measured in (an OWL instance)
optional	wasParsedFrom	Facebook_Post_1	The source that the information was parsed from, e.g. a sioc:Post (an OWL instance)
optional	wasResultOf ParsingActivity	Parsing_Activity _1	The sosens:ParsingActivity that transformed the above into a ssn:Observation (an OWL instance)

## USE CASE #1 Example

Two observations to be converted using the SOSENS API:

**1 Hardware Sensor Observation:** On 20/5/2015, 14:30, Digtherm\_1 measured the temperature of Area\_1 to be 50 Celsius.

**1 Social Sensor Observation:** On 20/5/2015, 14:30, Person\_1 made a post on Facebook (Facebook\_Post\_1) saying that Area\_1 was “warm”, which was picked up & parsed by a SOSENS Parser (Parsing\_Activity\_1)

## INPUT (JSON String)

```
{ "data": [{
  "observedBy": "Digtherm_1",
  "feature": "Area_1",
  "value": "50",
  "valueDataType": "integer",
  "valueType": "quantity",
  "property": "temperature",
  "unitOfMeasurement": "Celsius",
  "time": "2015-05-20T14:30:00"
},
{
  "observedBy": "Person_1",
  "feature": "Area_1",
  "value": "Warm",
  "valueDataType": "string",
  "valueType": "quality",
  "property": "temperature",
  "time": "2015-05-20T14:30:00",
  "wasParsedFrom": "Facebook_Post_1",
  "wasResultOfParsingActivity": "Parsing_Activity_2"
}
]}
```

*JSON*

## OUTPUT (RDF)

*RDF/Turtle*

```

:Digtherm_1_observation_temperature_2015_05_20_14_30_00
  a ssn:Observation ;
  ssn:featureOfInterest :Area_1 ;
  ssn:observationResultTime :instant_2015_05_20_14_30_00 ;
  ssn:observedBy :Digtherm_1 ;
  ssn:observedProperty :temperature ;
  ssn:sensorOutput :Digtherm_1_sOut_temperature_2015_05_20_14_30_00 .

:Person_1_observation_temperature_2015_05_20_14_30_00
  a ssn:Observation ;
  sosens:wasParsedFrom :Facebook_Post_1 ;
  sosens:wasResultOfParsingActivity :Parsing_Activity_1 ;
  ssn:featureOfInterest :Area_1 ;
  ssn:observationResultTime :instant_2015_05_20_14_30_00 ;
  ssn:observedBy :Person_1 ;
  ssn:observedProperty :temperature ;
  ssn:sensorOutput :Person_1_sOut_temperature_2015_05_20_14_30_00 .

:Digtherm_1_observationValue_temperature_2015_05_20_14_30_00
  a ssn:ObservationValue ;
  sosens:hasQuantityUnitOfMeasurement :Celsius ;
  sosens:hasQuantityValue "50.0"^^xsd:float .

:Digtherm_1_sensorOutput_temperature_2015_05_20_14_30_00
  a ssn:SensorOutput ;
  ssn:hasValue :Digtherm_1_oVal_temperature_2015_05_20_14_30_00 .

:Person_1_observationValue_temperature_2015_05_20_14_30_00
  a ssn:ObservationValue ;
  sosens:hasQualityValue "Warm"^^xsd:string .

:Person_1_sOut_temperature_2015_05_20_14_30_00
  a ssn:SensorOutput ;
  ssn:hasValue :Person_1_oVal_temperature_2015_05_20_14_30_00 .

:instant_2015_05_20_14_30_00 a time:instant ;
  time:inXSDDateTime "2015-05-20T14:30:00"^^xsd:dateTime .

```

## USE CASE #2: EXPERT Mode

### GLOBAL Options

You may override the `outputFormat` and `namespace` options of the API, in order to use your chosen RDF serialization and namespace, respectively

JSON: global			
type	variable	example	comment
optional	<code>outputFormat</code>	<code>n3</code>	The RDF serialization that will be used. Defaults to <code>n3</code> . Other options are <code>nt</code> , <code>pretty-xml</code> , <code>trix</code> , <code>turtle</code> , <code>xml</code>
optional	<code>namespace</code>	<code>http://example.org/data#</code>	The local namespace that will be used if URIs are not full http

### Replacing the default `ssn:Observation`, `ssn:SensorOutput` & `ssn:ObservationValue` classes

In practice, it can be very helpful to design customized versions (subclasses) of the `ssn:Observation`, `ssn:SensorOutput` and/or `ssn:ObservationValue` in order to leverage the benefits of OWL restrictions. For example:

- Customise the observation class
  - e.g. to make all observation values observe the same property (temperature), create a `:TemperatureObservation` subclass with the restriction:
    - `ssn:observedProperty value :temperature`
- Customise the `ssn:SensorOutput` class
  - e.g. to add restrictions on who can produce these outputs, create a `:TemperatureSensorOutput` class with the restriction:
    - `ssn:isProducedBy only :Thermometer`
    - `ssn:hasValue only :TemperatureValue`
- Customise the `ssn:ObservationValue` class
  - e.g. to make all value instances have the same unit of measurement, create a `:CelsiusValue` subclass with the restriction:
    - `:hasQuantityUnitOfMeasurement value :Celsius`

These customisations are, naturally, APPLICATION-specific. The SOSENS API allows declaring the various instances produced as members these customized classes by using the variables presented below, but writing the actual restrictions is left to the developer of each implementation.

JSON: data			
type	variable	example	comment
optional	observationClass	TemperatureObservation	The class the observation instance will be a member of (must be a subclass of <code>ssn:Observation</code> ). You may skip defining the “observedBy” variable if you declare this
optional	sensorOutputClass	TemperatureSensorOutput	The class the sensor output instance will be a member of (must be a subclass of <code>ssn:SensorOutput</code> )
optional	observationValueClass	CelsiusValue	The class the observation value instance will be a member of (must be a subclass of <code>ssn:ObservationValue</code> )

### Declaring URIs for any instance instead of default SOSENS URIs

If the “default” way of dealing with sensor observations in SSN is not fit-for-purpose, the SOSENS API allows you to declare your own instances as the predicates of the most common SSN properties. You may then structure these instances however you like. To fill in URIs, enter a string for using the local namespace (see GLOBAL options) or a full URI with `http://` for external namespace

JSON: data			
type	variable	example	comment
optional	observationURI	Observation_1	The URI of the observation instance
optional	sensorOutputURI	SensorOutput_1	The URI of the sensor output instance
optional	observationValueURI	ObservationValue_1	The URI of the observation value instance
optional	timeInstanceURI	TimeInstance_1	The URI of the instance that will be defined as the predicate of the <code>ssn:observationResultTime</code> property

