Supplementary materials

Teresa Zawadzka, Wojciech Waloszek

September 2021

1 Inconsistency Dataset for ROAD

To model the data set in ROAD the following steps were taken:

1. Instance of *Experiment* - *exp* individual represents an experiment.
2. Ten participants are represented as instances *P01*, ..., *P10* of *Participant* concept and *P01state*, ..., *P10state* of *ParticipantState* concept. Each pair of individuals representing participant and his/her state is related via the role *hasParticipant*. For each participant the sex is defined and the appearance in *ApperanceOcclusionModel* according to values presented in Table 1.
3. One activity representing the set of tasks done in the settled order is represented as *act* instance of *IndividualActivity* concept. There are tenth instances *actExecP01*, ..., *actExecP10* of *ActivityExecution* concept representing execution of activity for each participant. Each of these individuals is connected with *exp* instance via *hasScenario* role.
4. According to the fact that activities are individual for each participant one instance of *Participation* concept is defined (*partP01*, ..., *partP10*). Each of these individuals is connected with appropriate participant state and activity execution with *hasActivityExecution* and *hasParticipantState* roles (e.g. *hasActivityExecution*(*partP01*, *actExecP01*), *hasParticipantState*(*partP01*, *P01state*))
5. There are forty registered data. Each participant is recorded forth times (four cameras are installed) as *RegisteredData* concept instances: *rdP01DL*, ..., *rdP10DL*, *rdP01DR*, ..., *rdP10DR*, *rdP01UL*, ... *rdP10UL*, *rdP01UR*, ... *rdP10UR*.
6. There are forty *RegisteredChannel* instances corresponding to the registered data *rcP01DL*, ..., *rcP10DL*, *rcP01DR*, ..., *rcP10DR*, *rcP01UL*, ... *rcP10UL*, *rcP01UR*, ... *rcP10UR*. Each registered channel is connected with corresponding registered data and audio channel (*channelAudio*) with *hasRegisteredData* and *hasChannel* roles (e.g. *hasRegisteredData*(*rcP01*, *rdP01*). *hasChannel*(*rcP01*, *channelAudio*))

7. There are forty *Recording* instances corresponding to the registered channel *recP01DL*, ..., *recP10DL*, *recP01DR*, ..., *recP10DR*, *recP01UL*, ... *recP10UL*, *recP01UR*, ... *recP10UR*. Each recording is connected with corresponding registered channel and participation via *hasRegisteredChannel* and *hasParticipation* roles (e.g *hasRegisteredChannel*(*recP01UL*, *rcP01UL*), *hasParticipation*(*recP01UL*, *partP01*). *Recording* concept is subclass of *PropertyConcept* and key-value pair is used to define camera location property for each recording. Four new instances of *Property* concept are defined *camLocDL*, *camLocDR*, *camLocUL*, and *camLocUR*. Each of these instances has *hasKey* data type property set to "cameraLocation" (e.g. *hasKey*(*camLocDL*, "cameraLocation") and *hasValue* data type property set to "DL", "DR", "UL" or "UR" respectively: *hasValue*(*camLocDL*, "DL"), *hasValue*(*camLocDR*, "DR"), *hasValue*(*camLocUL*, "UL") and *hasValue*(*camLocUR*, "UR"). For each recording the appropriate *hasProperty* instance is defined, e.g. *hasProperty*(*recP01UL*, *camLocUL*).

8. There are also forty instances of *ObservableInformation* concept corresponding to appropriate recordings *obsInfP01DL*, ..., *obsInfP10DL*, *obsInfP01DR*, ..., *obsInfP10DR*, *obsInfP01UL*, ... *obsInfP10UL*, *obsInfP01UR*, ... *obsInfP10UR*. Each observable information is related to the corresponding recording via *hasRecording* role e.g. *hasRecording*(*obsInfP01DL*, *recP01DL*). Each observable information also points at facial expressions modality (individual *modalityFacialExpressions*) via *hasModality* role and movement life activity (individual *lifeActivityMovement*) via *hasLifeActivity* role.

9. Nine instances of *Measure* concept are defined: *valenceMeasure*, *arousalMeasure*, *neutralMeasure*, *happinessMeasure*, *angerMeasure*, *sadnessMeasure*, *disgustMeasure*, *fearMeasure*, and *surpriseMeasure*. Each of these instances has *type* set to float. The *range* of each measure but *valenceMeasure* is $< 0, 1 >$. For *valenceMeasure* range is set to $< -1, 1 >$. Individuals *valenceMeasure* and *arousalMeasure* point at measure names defined in *PADModeMeasureName* via *hasMeasureName* role. Analogically, individuals representing measures for Ekman's basic emotions point at measure names defined in *EkmanModelMeasureName*. For indvidual *neutralMeasure* object of *haMeasureName* role is set to *neutralState*.

10. Each observable information is a source of information for the set of 19 time series (9 generated by Face Reader, 6 by Express Engine and 7 by Luxand). For example for *obsInfP01DL* observation for FR: *P01DLFRvalence*, *P01DLFRarousal*,
P01DLFRneutral, *P01DLFRhappiness*, *P01DLanger*,
P01DLFRsadness, *P01DLFRdisgust*, *P01DLFRfear*
and *P01DLFRsurprise*. Each of the time series points to the corresponding measure. To model the system recognizing emotions again the inheritance from *PropertyConcept* is used. Analogically as for camera location three new instances of *Property* concept are defined: *sysFR*, *sysXP*,

sysLU. Each of these instances has *hasKey* data type property set to "system" (e.g. *hasKey*(*sysFR*, "system") and *hasValue* data type property set to "FR", "XP" or "LU" respectively: *hasValue*(*sysFR*, "FR"), *hasValue*(*sysXP*, "XP") and *hasValue*(*sysLU*, "LU"). Each time series is a subject of *hasProperty* role e.g. *hasProperty*(*P01DLFRvalence*, *sysFR*).

Table 1: Participants' characteristics.

participant	**sex**	**beard**	**moustache**	**glasses**
P01	Female	None	None	No
P02	Male	Heavy	Some	No
P03	Male	Some	Some	Yes
P04	Male	None	None	Yes
P05	Female	None	None	No
P06	Female	None	None	No
P07	Female	None	None	No
P08	Male	Some	Some	No
P09	Male	Heavy	Some	Yes
P10	Male	None	None	Yes