Towards data exchange formats for learning experiences in manufacturing workplaces

Fridolin Wild[1], Peter Scott[1], Paul Lefrere[1], Jaakko Karjalainen[2], Kaj Helin[2], Ambjoern Naeve[3], Erik Isaksson[3]

[1] The Open University, Knowledge Media Institute, Milton Keynes, UK
{fridolin.wild, peter.scott}@open.ac.uk
[2] VTT Technical Research Centre of Finland,
Human factors in complex systems, Tampere, Finland
{jaakko.karjalainen, kaj.helin}@vtt.fi
[3] Royal Institute of Technology (KTH), Stockholm, Sweden
amb@csc.kth.se

Abstract. Manufacturing industries are currently transforming, most notably through the introduction of advanced machinery and increasing degrees of automation. This has caused a shift in skills required, calling for a skills gap to be filled. Learning technology needs to embrace this change and with this contribution, we propose a process model for learning by experience to understand and explain learning under these changed conditions. To put this process into practice, we propose two interchange formats for capturing, sharing, and re-enacting pervasive learning activities and for describing workplaces with involved things, persons, places, devices, apps, and their set-up.

Keywords: Experience sharing, activity model, workplace model, awareness, augmented reality.

1 Introduction

The European (and global) manufacturing industry is currently undergoing significant transformation and will continue to change over the coming years. Intrinsically, the increasing presence and ability of robots and advanced machinery in production lines with their enhanced senses and increased dexterity (Frey & Osborne, 2013, p.38) are what triggers this shift, bringing along rising degrees of computerisation of jobs (ditto) and allowing for delocalisation of production.

Extrinsically, this transformation has started to cause a significant skills gap in the EU (and globally) with – on the one side – the highest overall unemployment rates observed in more than a decade (Eurostats, 2014; EC, 2013c, p. 2), especially amongst young people (EC, 2013b; EC, 2013d), and an ever increasing risk of redundancy for low and medium skilled workers in production.

On the other hand, several hundred thousand jobs in the EU remain unfilled, as there is a shortage in highly skilled personnel in manufacturing (EC, 2013a, p.10). Forecasts predict that this skills gap is likely to widen in coming years up to 2020

(McKinsey, 2012, p. 45). In fact, manufacturing is currently one of the three sectors hit most hard by this skills shortage in the EU (EC, 2013c, p. 5). Formal secondary and tertiary education haven't managed to create and won't succeed in producing the supply required, neither in numbers, nor with respect to matching skills profiles. Moreover, high attrition rates in education have further eroded the foundation.

Technology enhanced learning has the potential to play an important role in overcoming this existing skills gap in manufacturing – when applied effectively and when motivating the development of competences in key areas required through the capturing and re-enactment of learning activities.

Within this contribution, we first define a learning process model that is capable of integrating classical (learning content oriented) and novel pervasive (Augmented Reality and Internet of Things oriented) elements in learning at manufacturing workplaces. From there, we introduce a proposal for an activity modelling language (activityML) for representing activity descriptions required in augmented reality enabled learning experiences. Moreover and in section 4, we introduce the needed workplace modelling language (workplaceML), which can be used to describe the tangibles (things, places, persons), configurables (apps, devices), and triggers (detectables, overlays) of a particular workplace. We relate our work to precursors in Section 5 to then wrap up the paper with an outlook and open research challenges.

2 Process model of learning by experience

New skills for new jobs not only demand an enhancement of the deep professional skills to achieve a 'master level of performance', but also necessitate development and upgrading of competence to innovate, for lifelong learning, and for learning through social interaction (Wild et al., 2013, p. 12f).

Achieving a master level of performance and developing competence to innovate in the sense of building up "the ability to generate ideas and artefacts that are new, influential, and valuable" (FET, 2011) are – at least in manufacturing and at least for small and medium enterprises – very closely intertwined.

	Tacit	**Explicit**
Tacit	*Socialisation*	*Externalisation*
Explicit	*Internalisation*	*Combination*

Table 1. Knowledge conversion modalities.

Similarly, the other two, namely lifelong learning and social learning competence, both pay tribute to the observation that "people carry and create knowledge" and that "any company knowledge management strategy must rely primarily on people, and support [of] the knowledge creation chain" (Krassi and Kiviranta, 2013, p.29). Both of them aim to facilitate "bi-directional tacit-explicit knowledge conversion" (Nonaka, 1994, p.19) along the four modalities listed in **Table 1**: externalisation (tacit-to-explicit), internalisation (explicit-to-tacit), socialisation (tacit-to-tacit), and combination (explicit-to-explicit).

While 'competences' are typically defined to subsume knowledge, skills and other abilities, in the context of manufacturing – as the word already suggests – motoric and artistic skills require special attention. Kinaesthetic learning elements relate in manufacturing environments to controlling own body movement and handling objects skilfully and timely (cf. Gardner, 1984: bodily-kinaesthetic intelligence).

With the rise of Wearable Computing, the Internet of Things, and Augmented Reality, capturing and observing kinaesthetic performance becomes possible in a fundamentally different way, as, for example, pioneered in the fitness and health sector.

Reflective learning processes that cater for kinaesthetic and non-kinaesthetic elements can be broken down into five distinct process steps: enquire, mix, experience, match, optimise (Wild et al., 2013, p.29ff). The steps do not necessarily prescribe a single route and order, in which they should be taken, but are interconnected as indicated in **Fig. 1**: it is a cyclical model with built-in support for experience tracking, analytics, and guidance, supporting flexible mixes and dynamic optimization for on- and off-the-job workplace learning.

Fig. 1 depicts the individual steps of this process model: at its core, blue-collar workers *experience* learning in an episodic way (on and off the job). Experiencing thereby relates to both re-enacting explicit learning activities as well as engaging in open innovation activities.

Experiencing learning tightly interacts with *enquiry*: whenever novel needs arise or (wicked) problems are encountered on the job, the enquiry step supports the user in identifying relevant learning opportunities (such as gaps in knowledge, new learning opportunities arising, etc.). In parts, this relates to navigational positioning support in the workplace reference space's skills taxonomy to clearly determine the competence sought after. This, however, also relates to supporting discovery beyond existing or – particularly relevant for SMEs – so-far tacit knowledge.

Through tracking of experiences made, potential competence gaps (ignorance) can be uncovered, uncovering thereby either supported by the system in the *matching* step (see below; aiming to help unveil shortcomings the user is unaware of) or – proactively, where awareness is given – through user enquiry.

Once needs or problems are identified, the *mixing* step comes into play: here, the learner is supported in selecting relevant existing mixes or creating new and adapting existing mixes. While standard problems have standard solutions, smart factories enable their workers to rapidly compile mixes that satisfy needs, but at the same time ensure documentation of knowledge, where it is created. Such mix is essentially a serialized, activity-focused representation of the specific workplace and the jobs to be enacted within it, instantiating an abstract workspace to a level of concreteness where actions are named, locations resolved, and objects as well as tools uniquely identified.

Moreover, the activity mix models validation constraints, by which the *matching* step can determine whether there is evidence that the user actually performed the action steps as required. Constraints model learning flows including exception handling. The constraint-matching step picks up on strategic performance indicators and their defined tolerance boundaries set at design time, and connects them to the observed operational performance as tracked by the *experience* step. Reports for performance analytics can be generated live, condensing performance records (from an xAPI endpoint; ADL, 2013) into comprehensive reports, potentially contrasting performance of individuals with (de-identified) benchmarks.

Optimisations then take such analytics data and performance benchmarks to recommend repetition, alternative resources, or even a change of path.

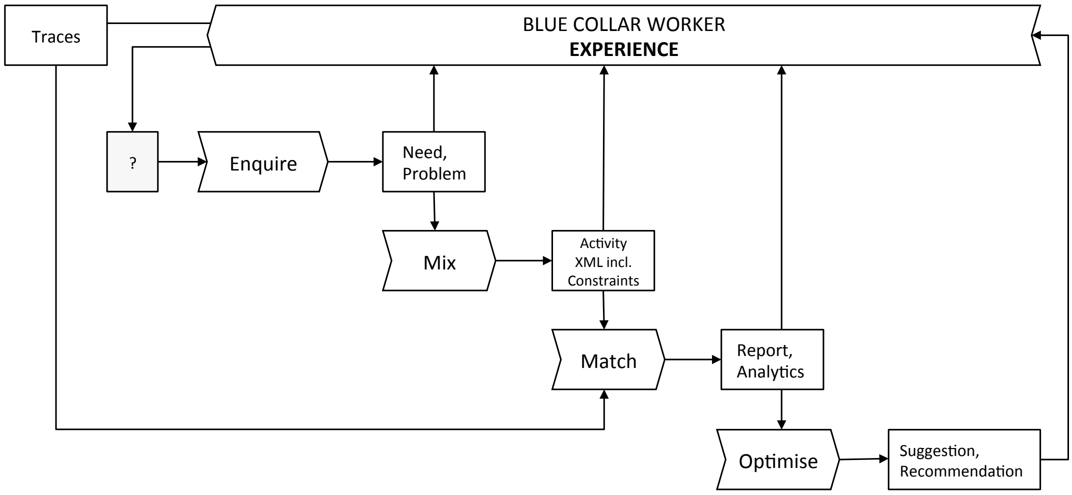

Fig. 1. Process model for learning by experience.

The bi-directional conversion between tacit and explicit knowledge is modelled in the process loops between the big process step ‘experiencing’ and the smaller ones ‘enquiry’, ‘mix’, ‘match’, and ‘optimise’: explication converts tacit knowledge to explicit as indicated by the outputs ‘traces’, ‘needs’, ‘activity mixes’, ‘analytics reports’, and ‘recommendations’. When such outputs are used to scaffold a learning experience, they are internalised. Remixing combines existing knowledge, and tracking and evidence recording helps with converting tacit to explicit knowledge. Moreover, activity mixes can involve socialisation and social sharing.

3 Modelling activities

A common representation format is key requirement for an efficient exchange of activity mixes. What makes it particularly challenging to define activity mixes in a process for learning by experience is that it requires not only orchestrating user interaction across multiple devices within a single activity, but also integrating the tracking of and reacting to user interaction across these devices and – even more so – their different sensors. Validating that the user actually did something (like moving to a particular location or like picking up a particular object in the real world), requires specifying validation constraints that can be checked and that express *which* soft- and hardware sensors have to pick up on *what* user (or app) behaviour. For this we propose activityML (activity modelling language), an XML dialect.

Fig. 2 provides an overview on the conceptual model of activityML. **Fig. 3** adds an example activityML file. The root node is 'activity'. Each activity needs to specify the URL of the workplace description file, a name, and the language (in addition to the unique activity id). The activity is then broken down into 'action' steps, each of them being a self-contained unit, describing 'summons' for action chaining, 'constraints' for action validation, and 'messages' for communication, as well as 'instruction' to be shown or the 'app' (widget or app) to be launched.

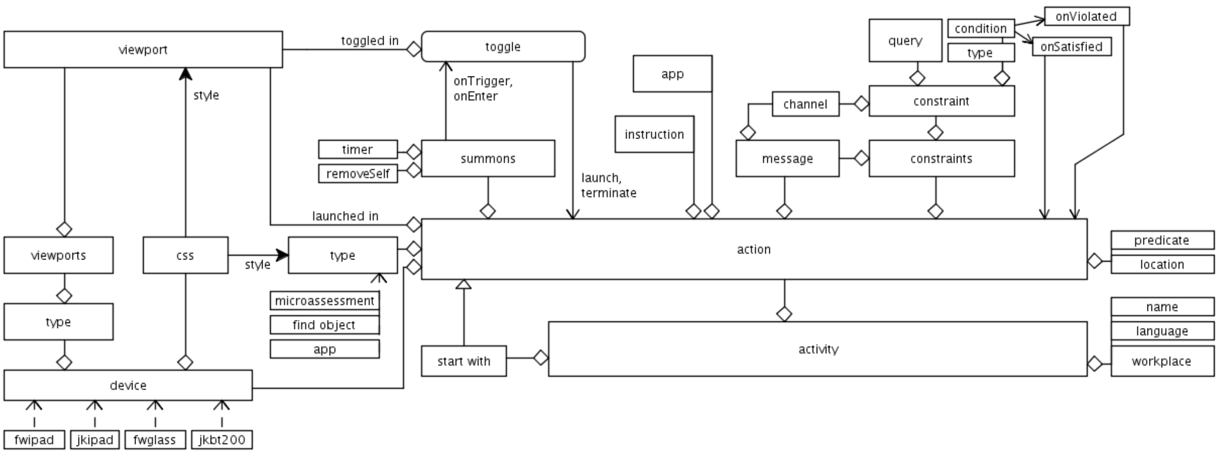

Fig. 2. Conceptual model of the activityML.

Moreover, styling information is linked using cascading style sheets over the action 'type' and the 'device' and its 'viewports' defined. Currently, there are three viewports defined: 'objects', 'actions', and 'reactions'. They refer to particular areas of the screen reserved for inserting actions and the related display data.

Each 'action' has a 'predicate', which is the verb required for inserting trace statements to the xAPI (ADL, 2013) tracking endpoint. Each action can optionally specify a 'location', i.e. a defined 'place' of the workplace model, in which it shall happen.

Chaining of actions is modeled through specifying for each action, which other actions it 'summons' – either when launching the action ('onEnter') or when events are triggered ('onTrigger'). The Boolean 'removeSelf' decides on whether the action is removed from the viewport or sustained when the summons are executed. A timer can be set to automatically trigger summons after a given interval (in milliseconds). Summoning an action twice will first show, then remove it from the viewport (hence 'toggle'). Each summoned toggle specifies the 'viewport' in which it is toggling an action. The summons are also used to activate in a context-dependent way the ids of actions relevant for the next step. By using the id of a tangible (e.g. thing or person) as the id of an action step, for example, an object detection engine can trigger the launching (or termination) of the according action.

The ‘constraints’ define, how the system can validate that certain user interaction and other observable conditions are given the way they were modeled. For example, as depicted in **Fig. 3**, a constraint of type ‘onEnter’ can be defined that checks whether the learner has certain basic ICT skills. Constraints are specified in a given query language (in the example: SQL) and they define their own action branching for ‘onSatisfied’ and ‘onViolated’ conditions.

To enable communication between devices and to allow for communicating with overlays (as specified in the workplace model), ‘messages’ can be used: each ‘message’ specifies, which ‘target’ (device, thing, person, …) and ‘id’ they want to communicate with. In case that the message is to a thing, the ‘overlay’ needs to be specified. Messages can also declare explicitly, which communication ‘channel’ they want to use (e.g. a real-time presence channel ‘rpc’ or an ‘xapi’ endpoint).

```xml
<?xml version="1.0" encoding="utf-8"?>
<activity

    id="artel14"
    name="do-torial: learn about something complex"
    language="english"
    workplace="http://tellme.kmi.open.ac.uk/tellmar/development/data/workplace.php?id=kmi"

    start="welcome"

    >

    <action id="welcome" device="ipadba44ff" viewport="actions" predicate="find" location="SprayBooth">
        <summons type="onEnter" removeSelf="true" timer="500">
            <toggle viewport="actions" type="action">findOrderSheet</toggle>
        </summons>
        <constraints type="onEnter" satisfied="" violated="error">
            <constraint type="boolean" condition="true">(SELECT count(skill) FROM userskills WHERE user=%u% AND skill='basicICT')>0</constraint>
        </constraints>
        <instruction><![CDATA[ <h1>Check the Manufacturing Order!</h1><p> ... </p> ]]></instruction>
    </action>

    <action id="error" device="ipadba44ff" viewport="actions" predicate="error">
        <summons type="onExit" removeSelf="true">
            <toggle viewport="actions" type="action">welcome</toggle>
        </summons>
        <instruction><![CDATA[ <h1>Error!</h1><p>You don't fulfill the preconditions for this learning job! Tap to try again.</p> ]]></instruction>
    </action>

    <action id="findOrderSheet" device="ipadba44ff" viewport="actions" predicate="find" location="SprayBooth">
        <messages>
            <message target="thing" id="orderSheet" overlay="label1">New job for Fridolin</message>
            <message target="device" id="fwbt200" channel="rpc">Ping ;)</message>
        </messages>
        <summons type="onEnter" removeSelf="false">
            <toggle viewport="actions" type="action">action15</toggle>
        </summons>
    </action>

    <action id="action15" device="fwbt200" viewport="actions" predicate="find" location="SprayBooth">
        <messages>
            <message channel="rpc" device="ipadba44ff">received action15 on moverio bt-200 glasses</message>
        </messages>
        <summons type="onTrigger" removeSelf="true">
            <toggle viewport="objects" type="object" device="ipadba44ff">orderSheet</toggle>
            <toggle viewport="actions" type="action">openSmartPlayerWidget</toggle>
        </summons>
        <instruction><![CDATA[ <h1>Go find the order sheet!</h1><p>The order sheet lists your next job...</p> ]]></instruction>
    </action>

    <action id="openSmartPlayerWidget" device="ipadba44ff" viewport="actions" predicate="watch">
        <app type="SmartPlayer" id="16" />
        <summons type="onExit" removeSelf="true">
            <toggle viewport="actions" type="action">interpretMO</toggle>
            <toggle viewport="actions" type="action">checkWorkplace</toggle>
        </summons>
    </action>

</activity>
```

Fig. 3. Example activityML code.

Fig. 3 provides an example mock activity model. In this code example, the activity is broken down into five action steps, four of which are to be executed on a tablet PC, while one will be launched on a pair of augmented-reality (see-through) glasses. The user interaction starts with a welcome instruction to check the manufacturing order (auto-removed after 500ms) in which a constraint validation is performed checking whether the user has the required skills (from a user profile).

If this constraint is validated, the user proceeds to finding the order sheet; otherwise an error message is displayed. To support finding the order sheet object, an action is launched on the glasses, which is toggled with sending the according id ('action15'). Once the object has been found in the viewfinder of the glasses, the next action – playing multimedia instructions in the smart player app – follows, this now again on the tablet PC.

4 Modelling workplaces

To create interoperability of applications interpreting activityML, a description of the workplace is required in a defined interchange format. We propose for this workplaceML, an XML dialect to describe the tangibles, configurables, and basic triggers of a workplace. The 'tangibles' thereby refer to 'things', 'places', and 'persons', see **Fig. 4**. The 'configurables' fall into two classes, namely 'devices' and 'apps'.

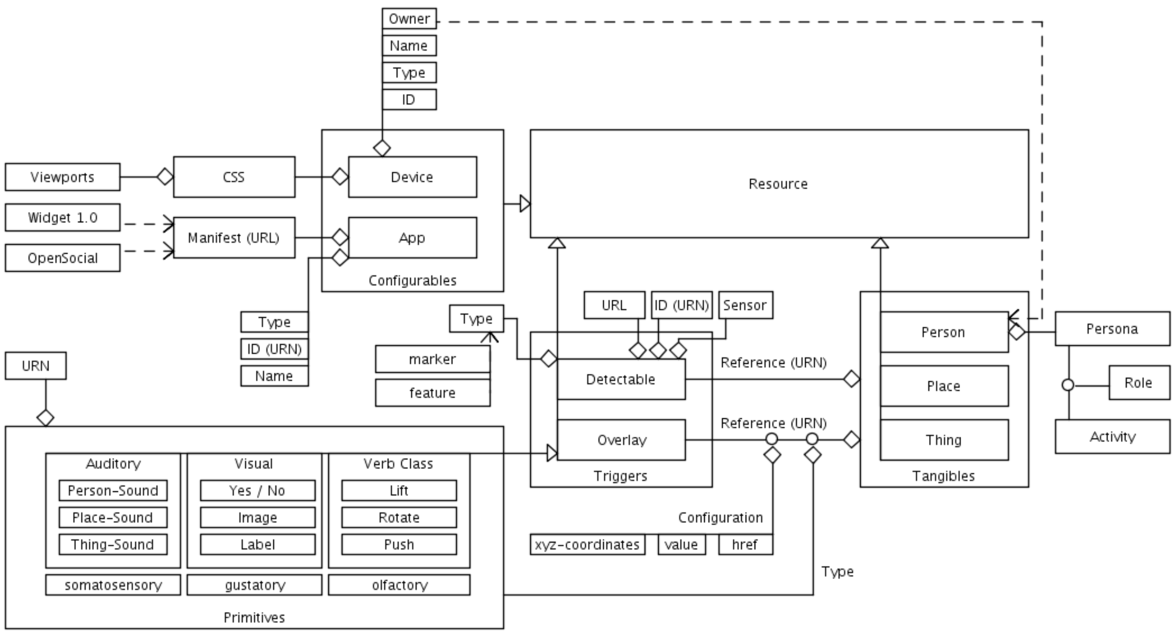

Fig. 4. Conceptual model of workplaceML.

Finally, the 'triggers' group together both 'detectables' (such as markers) and the primitives of 'overlays'. The relationship between tangibles and triggers is crucial: each tangible can specify the corresponding 'detectable' to determine how it can be detected: it can name the id of a marker, the id of a feature cloud, or even the id send by an Internet of Things component such as an intelligent toolbox that monitors through sensors which tools are taken out or put back in. Moreover, each tangible can list the overlay primitives supported and configure them if required. For example, a 'YesNo' visual overlay does not require additional configuration: an app identifying the tangible will automatically overlay a green circle when it is relevant to the current action step (and a red cross, when not). This is different, for example, for an image overlay primitive: in that case, the tangible needs to specify via 'src' the path to the image to be displayed (and whether it shall be anchored to the detectable or to the horizon).

```xml
<?xml version="1.0" encoding="utf-8"?>

<workplace

    id="kmi"
    name="Knowledge Media Institute of The Open University"

    >

    <resources>

        <tangibles>

            <things>
                <thing id="thehammer1" name="The Hammer" urn="/tellme/object/hammer" detectable="010">
                    <overlay id="YesNo" enabled="true" x-offset="0" y-offset="0" z-offset="0" />
                    <overlay id="image" x-offset="0" y-offset="0" z-offset="0" src="images/hammer.png" />
                    <overlay id="image" anchored="marker" src="images/currentPositionLine.png" />
                    <overlay id="image" anchored="world" rad="0degree" src="images/currentPositionLine.png" />
                    <overlay id="rotate" />
                    <overlay id="label" enabled="true" name="Head of the Hammer" x-offset="0" y-offset="0" z-offset="0" />
                </thing>
            <things>

            <places>
                <place id="spraybooth" name="The Spray Booth" detectable="005" type="/tellme/location/spraybooth" />
                <place id="platform" name="The Spray Booth" detectable="006" type="/tellme/location/spraybooth">
                    <location lat="12.3121222" long="123.3422222" radius="0.000003"/>
                </place>
            </places>

            <persons>
                <person id="fridolin" name="Fridolin Wild" mbox="f.wild@open.ac.uk" detectable="002" persona="/tellme/user/learner" />
                <person id="jake" name="Jake" detectable="003" persona="/tellme/user/trainer">
                    <device "bacefd33ba" volume="120%" />
                </person>
            </persons>

        </tangibles>

        <configurables>

            <devices>
                <device id="bacefd33ba" type="ipad" name="Fridolin's iPad" owner="fridolin" />
                <device id="22ffdab321" type="moveriobt200" name="Fridolin's Glasses" owner="fridolin" />
                <device id="bac3dddfff" type="ipad" name="Jake's iPad" owner="jake" />
            </devices>

            <apps>
                <app type="widget" id="smartplayer" name="TELLME smart player widget" manifest="http://demos.piksel.com/smartplayer.html" />
                <app type="widget" id="rating" name="TELLME.live rating widget" manifest="http://tellme.kmi.open.ac.uk/widgets/rating.php" />
                <app type="widget" id="precisionteaching" name="PT assessment" manifest="http://tellme.kmi.open.ac.uk/widgets/pt.php" />
                <app type="app" id="flashmeeting" name="fm.mobile" call="itc://fmMobile.app" />
            </apps>

        </configurables>

        <triggers>

            <detectables>
                <detectable id="010" sensor="engine" type="marker" />
                <detectable id="015" sensor="engine" type="feature" file="featureCloud015.fcl" />
                <detectable id="020" sensor="snapon" type="IoT" event="returnedPliers"/>
            </detectables>

            <overlays>
                <primitive id="YesNo" type="visual" />
                <primitive id="image" type="visual" />
                <primitive id="label" type="visual" />

                <primitive id="person-sound" type="auditory" file="person-sound.wav" />
                <primitive id="thing-sound" type="auditory" file="person-sound.wav" />
                <primitive id="place-sound" type="auditory" file="person-sound.wav" />

                <primitive id="lift" type="motionverb" file="animated-arrow-away.fbx" />
                <primitive id="rotate" type="motionverb" file="animated-arrow-clockwise.fbx" />

                <primitive id="3dinstructible" type="three.js" file="http://threejs.org/examples/webgl_geometry_cube.html" />
            </overlays>

        </triggers>

    </resources>

</workplace>
```

Fig. 5. Example of a workplaceML file.

Certain verbs of handling and movement can be predicates of an action step (e.g. ‘lift’ or ‘rotate’) and they overlay primitives can be enabled accordingly in the configuration of the tangible.

The ‘configurables’ specify for each device the ‘owner’, a human-readable ‘name’, the ‘type’ (e.g. ‘iPad mini’ versus ‘google glass’) and its unique id. This ensures that messages can be delivered and actions can be launched and styled correctly. The ‘apps’ define the URL of the manifest file of e.g. a ‘widget 1.0‘ compliant or ‘OpenSocial’ compliant widget.

The code example presented in **Fig. 5** now provides the required workplace information on the tangibles (places, things, persons), configurables (apps, devices), and triggers (detectables, overlays).

From the bottom to the top, first the overlay primitives are described, i.e. a generic definition of which types of overlays exist and which modality they are overlaid in. For example, there is a person sound and there is an image overlay.

Next, the detectables are defined: this enables a pre-trained marker (‘010’), a featureless object model (‘015’), and an event from an Internet of Things sensor (‘020’). There are two types of configurables defined in this example: the devices (e.g. of type ‘ipad’ or ‘moveriobt200’) as well as the apps that can be launched, some of which through calls to the device app to be launched, others as html5 widgets.

Finally, definitions of persons, places, and things follow. Here, each tangible can further configure the overlay primitives described at the very end of the script. For example, the thing ‘thehammer1’ is bound to the marker ‘010’ and configured to support image overlays using a picture of the hammer and setting the xyz-offsets as required.

5 Related work

In Naeve et al. (2014), we have presented generic, complementary deliberations about workplace models as well as an earlier, less elaborate version of the interchange format for activities proposed (p.48ff).

ACTIVITY-DL (Lanquepin, 2013; Barot et al., 2013) builds on the former HAWAI-DL proposal of the same group and provides a hierarchical way to describe tasks for virtual reality environments. While the task description is very advanced, the language lacks capabilities for device and multi-sensor integration. ACTIVITY-DL refers to its precursors MAD (Methode Analytique de Description; Sebillotte & Scapin, 1994) and GTA (Groupware Task Analysis; Veer et al., 1996), both focusing on analysing work tasks in interaction with user interfaces. While both provide conceptual insights (e.g. on timing and on condition modelling), they do not provide bindings against an interchange format.

6 Conclusion and outlook

In this contribution, we have rooted our motivation for creating the required exchange formats for capturing and sharing (kinaesthetic) learning experiences in manufacturing workplaces. The transformation the industry is currently undertaking has left a skills gap, which can be closed using learning technology apt to capture, share, and guide in re-enacting innovative production activity. For this, we have described the learning process and proposed two novel interchange formats for exchanging executable descriptions of learning by doing activity and workplaces. The exchange formats are implemented in the ARgh! prototype, a first glance of which is published in the proceedings of the main conference (Wild et al., 2014).

In a world, where the time required for updating must be significantly smaller than the half-life of knowledge documented, this becomes a key enabler for experience sharing and a corner stone for success.

The specifications have already been tested against a range of storyboards of the TELL-ME project and with participants of the joint European doctoral summer school in TEL (JTEL'14). The upcoming user pilots in the TELL-ME are expected to lead to further refinements. In particular, work is undergoing at the moment to further refine the predicate vocabulary and fine-tune it to the three pilot workplaces tested (aviation, furniture production, textile inspection and production). Moreover, the xAPI integration already feeds back to the constraint validation and further updates on query language and reasoning are to be expected.

Acknowledgements

The research leading to the results presented in this contribution has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318329 (the TELL-ME project, http://tellme-ip.eu). The authors would like to express their gratitude to the partners who have been involved in the related research work in TELL-ME.

References

1. Frey, C.; Osborne, M. (2013): The future of employment: how susceptible are jobs to computerisation?, unpublished manuscript, University of Oxford: Oxford Martin School.
2. European Commission (2013a): Annual Growth Survey 2012, COM(2011) 815 final.
3. European Commission (2013b): Europe 2020 Factsheet on Youth unemployment, online at: http://ec.europa.eu/europe2020/pdf/themes/21_youth_unemployment.pdf
4. European Commission (2013c): Draft joint employment report 2014 (JER), COM(2013) 801 final, online at: http://ec.europa.eu/europe2020/pdf/2014/jer2014_en.pdf
5. European Commission (2013d): Working together for Europe's young people: A call to action on youth unemployment. COM(2013) 447 final.
6. Eurostats (2014): Unemployment statistics, online at: http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Unemployment_statistics
7. McKinsey Global Institute (2012): The world at work: Jobs, pay, and skills for 3.5 billion people, June 2012, McKinsey.
8. Wild, Fridolin; Scott, Peter; Da Bormida, Giorgio; Lefrere, Paul; Naeve, Ambjoern; Isaksson, Erik; Valdina, Alessandro; Nina, Manuel; Marsh, Jesse (2013): Learning By Experience: The TELL-ME Methodology (phase one), deliverable d1.2, TELL-ME consortium.

9. FET (2011): Creativity and ICT, FET Consultation Workshop, Brussels, 28 November 2011. Report dated December 2011.
10. Krassi, Boris; Kiviranta, Sauli (2013): The unity of human and machine, In: VTT Impulse: Science, 2013(1): 24-29.
11. Nonaka, I. (1994) A Dynamic Theory of Organizational Knowledge Creation, Organization Science, Vol. 5, No. 1 (Feb., 1994), pp. 14-37.
12. Gardner, H. (1984): Frames of mind: the theory of multiple intelligences, Heinemann.
13. ADL (2013): Experience API, version 1.0.1, Advanced Distributed Learning (ADL) Initiative, U.S. Department of Defense.
14. Naeve, A.; Isaksson, E.; Lefrere, P.; Wild, F.; Tobiasson, H.; Walldius, Å; Lantz, A.; Viitaniemi, J.; Karjalainen, J.; Helin, K.; Nuñes, M.J., Martín, J. (2014): Integrated industrial workplace model reference implementation, deliverable d4.3, TELL-ME consortium.
15. Lanquepin, V.; Carpentier, K.; Lourdeaux, D.; Lhommet, M.; Barot, C.; Amokrane, K. (2013): HUMANS: a HUman Models based Artificial eNvironments Software platform, In: Proceedings of Laval Virtual VRIC'13, March 20-22, 2013 Laval, France, ACM.
16. Barot, C.; Lourdeaux, D.; Burkhardt, J.-M.; Amokrane, K.; Lenne, D. (2013): V3S: A Virtual Environment for Risk-Management Training Based on Human-Activity Models, In: Presence, Vol. 22, No. 1, Winter 2013, 1–19
17. Sebillotte, S.; Scapin, D. (1994): From users' task knowledge to high-level interface specification, International Journal of Human-Computer Interaction, 6:1, 1-15.
18. Veer, G.; Lenting, B.; Bergevoet, B. (1996): GTA: Groupware Task Analysis - Modeling Complexity, In: Acta Psychologica 91(1996):297-322.
19. Wild, F.; Scott, P.; Karjalainen, J.; Helin, K.; Lind-Kohvakka, S.; Naeve, A. (2014): An augmented reality job performance aid for kinaesthetic learning in manufacturing work places, In: Open Learning and Teaching in Educational Communities, Proceedings of EC-TEL 2014, Springer: Berlin.